Real-time 3D Morphable Shape Model Fitting to Monocular In-the-wild Videos

Patrik Huber

Submitted for the Degree of
Doctor of Philosophy
from the
University of Surrey

Centre for Vision, Speech and Signal Processing
Faculty of Engineering and Physical Sciences
University of Surrey
Guildford, Surrey, GU2 7XH
United Kingdom

June 2017

Abstract

Reconstructing 3D face shape from a single 2D photograph as well as from video is an inherently ill-posed problem with many ambiguities. One way to solve some of the ambiguities is using a 3D face model to aid the task. 3D Morphable Face Models (3DMMs) are amongst the state of the art methods for 3D face reconstruction, or so called 3D model *fitting*. However, current existing methods have severe limitations, and most of them have not been trialled on in-the-wild data. Current analysis-by-synthesis methods form complex non-linear optimisation processes, and optimisers often get stuck in local optima. Further, most existing methods are slow, requiring in the order of minutes to process one photograph.

This thesis presents an algorithm to reconstruct 3D face shape from a single image as well as from sets of images or video frames in real-time. We introduce a solution for linear fitting of a PCA shape identity model and expression blendshapes to 2D facial landmarks. To improve the accuracy of the shape, a fast face contour fitting algorithm is introduced. These different components of the algorithm are run in iteration, resulting in a fast, linear shape-to-landmarks fitting algorithm. The algorithm, specifically designed to fit to landmarks obtained from in-the-wild images, by tackling imaging conditions that occur in in-the-wild images like facial expressions and the mismatch of 2D–3D contour correspondences, achieves the shape reconstruction accuracy of much more complex, nonlinear state of the art methods, while being multiple orders of magnitudes faster.

Second, we address the problem of fitting to sets of multiple images of the same person, as well as monocular video sequences. We extend the proposed shape-to-landmarks fitting to multiple frames by using the knowledge that all images are from the same identity. To recover facial texture, the approach uses texture from the original images, instead of employing the often-used PCA albedo model of a 3DMM. We employ an algorithm that merges texture from multiple frames in real-time based on a weighting of each triangle of the reconstructed shape mesh.

Last, we make the proposed real-time 3D morphable face model fitting algorithm available as open-source software. In contrast to ubiquitous available 2D-based face models and code, there is a general lack of software for 3D morphable face model fitting, hindering a widespread adoption. The library thus constitutes a significant contribution to the community.

Keywords: 3D Morphable Face Models, 3D Face Reconstruction, Real-time, Open Source Software

Email: `p.huber@surrey.ac.uk`, `patrikhuber@gmail.com`
Web: `http://www.patrikhuber.ch`

Acknowledgements

I want to express my deepest gratitude and thank you to Josef, for his guidance, patience, for always making time, sharing his tremendous knowledge, and for his unconditional support. Thank you to Matthias and Bill for their encouragement, discussions and supervision, and thank you to Klaus, Frank and Thorsten for all their support and shared knowledge. Thank you to all of you for giving me the opportunity to do this PhD.

Thank you to Anna, Adam, Zhen-Hua and Philipp, for their love, friendship, collaborations, and the always fruitful discussions.

Thank you to Andreas and Pia for their love and support in all aspects. And thank you to Sabina for all her love during her life.

Contents

List of Symbols

$\mathcal{M}$	A PCA model with mean, eigenvalues and eigenvectors
$\mathbf{S}$	A vector representing a shape model instance
N	Number of mesh vertices
$\bar{\mathbf{v}}$	Shape model mean
$\boldsymbol{\sigma}$	Shape model eigenvalues
$\mathbf{V}$	Shape basis
$\boldsymbol{\alpha}$	A vector of shape coefficients
m	Number of shape principal components
$\mathbf{B}$	Expression blendshapes basis
$\boldsymbol{\psi}$	A vector of blendshape coefficients
k	Number of expression blendshapes
$\boldsymbol{\rho}$	A vector of camera parameters
$\mathbf{A}$	Affine camera matrix
$\mathbf{C}$	Orthographic camera matrix
$\mathbf{P}$	Block-diagonal stacked camera matrices

Acronyms

SFM, sfm	Surrey Face Model
3DMM	3D Morphable Model
ASM	Active Shape Model
AAM	Active Appearance Model
PCA	Principal Component Analysis
IMDR	Iterative Multi-resolution Dense 3D Registration
SVD	Singular Value Decomposition
SOP	Scaled orthographic projection
RGB	Red, green and blue colour values
MFF	The Multi-features fitting algorithm
MCMC	Markov chain Monte Carlo
KF-ITW	The Imperial College KF-ITW dataset
ITW	In-the-wild
3DDFA	The 3D Dense Face Alignment algorithm
ICP	Iterative Closest Point
CVSSP	Centre for Vision, Speech and Signal Processing

List of Figures

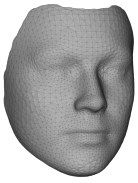
Chapter 1

Introduction

A MetraLabs SCITOS G5, used for human-robot interaction.

Face modelling, analysis and reconstruction are very prominent topics in computer vision, and have been widely researched over the last decades. There are a great many applications to these topics, ranging from image segmentation or reconstructive surgery in medical applications to virtual avatars, face recognition, and human-computer interaction. With augmented and virtual reality, robotics, and technology being and becoming prevalent in many aspects of everyday life, the need for robust techniques for face modelling, analysis and reconstruction will only increase even further in the near future.

Most often, a type of face model is used for face analysis and reconstruction. Up until a few years ago, 2D methods have dominated the research landscape for their feasibility with regards to creating them, their robustness and their speed, starting with Active Shape Models [CT92, CTCG95] and soon after Active Appearance Models [CET98, CET01]. Over the last 5 to 10 years, more research has focused on 3D methods, trying to overcome some of the limitations of 2D methods, perhaps starting with the seminal paper of Blanz & Vetter about 3D Morphable Models [BV99]. In the past few years, 3D methods are employed more widely in the field of face modelling and analysis, however, they have still not been proven feasible or effective for many tasks.



An instance of a 3D morphable face model.

In general, 3D models are harder to construct than 2D-based models. For example, to construct a 3D Morphable Model (3DMM), 3D scans of several hundred subjects are required, and the scans need to be preprocessed and brought into dense correspondence. For 2D models, there is usually an abundance of 2D images with labelled landmarks available, and that is all that is required. Also, recovering the 3D model parameters from an input 2D image (*model fitting*) is much more involved than fitting a 2D-based model, both in terms of complexity of the fitting algorithms, as well as run time. Thus, 2D methods are still prevalent in today's landscape. For example, a multitude of public implementations are available for ASM and AAM model building and fitting, but implementations of 3D face models and fitting are very hard to come by.

Many tasks in face analysis involve reconstructing a face in 3D from a single 2D photograph. Inferring 3D information from 2D data is an inherently ill-posed problem, and one needs to inject some form of prior knowledge to make the task feasible. Often, a 3D face model is used precisely for that purpose: such a face model is constructed from 3D face scans by modelling their statistical variability. This statistical model of faces is then used to aid the task of 3D face reconstruction from 2D data.

An attractive property of 3D face models and 3D morphable models in particular is that, in contrast to 2D methods, the pose of a face is clearly separated from the shape, and the model's projection to 2D is expressed with an explicit, physical camera model. This enables, for example, to compute 3D surface normals of a face mesh and parts that are self-occluded. Traditionally, 3DMMs also model skin albedo (appearance), which is captured from 3D scans, usually in lab-conditions. Together with an illumination model, for example Phong illumination, a full imaging model is defined, with which faces can be synthesised (rendered). The task of

model fitting to an image is then often approached by *analysis-by-synthesis*, which aims to fit the model by minimising the RGB reconstruction error of the synthesised model and the face in the original image. However, such an optimisation problem is highly nonlinear, with numerous parameters to estimate. Additionally, while fitting the albedo and illumination model has been shown to work well in controlled conditions, it has not yet been proven effective on in-the-wild images, which contain strong appearance variations, difficult illumination conditions, and expressions, amongst other things, making the optimisation problem very hard to solve. Traditionally, nonlinear optimisers like the Levenberg-Marquardt or L-BFGS algorithm are used, and most fitting algorithm include a term containing a landmark constraint into their cost function, and thus heavily depend on labelled or detected 2D facial landmarks. Apart from the difficulty of the optimisation, these nonlinear fitting methods are usually slow and have a run time in the order of minutes. For similar reasons, few work tackles 3DMM fitting to uncalibrated, monocular in-the-wild videos.

Recently, fitting algorithms decoupling the photometric (appearance and illumination) and geometric (pose and shape) parts have shown promise. In the geometric part, these algorithms mostly fit the shape to facial landmarks only. This is especially promising as 2D facial landmark detection is a quite mature research area, and facial landmarks can be detected robustly on in-the-wild images. However these 3DMM fitting algorithms have also not been applied to in-the-wild images and videos yet, and many of them do not contain crucial parts like a facial expression model and fitting, without which it is a futile effort to employ these algorithms on in-the-wild images.

In the light of the promising results of these landmark-fitting algorithms, and the robustness of facial landmark detection, in this thesis, we develop a real-time landmark-fitting algorithm for in-the-wild images and videos,

and investigate the shape reconstruction quality of shape-from-landmarks, compared with recent nonlinear and learning-based approaches.

1.1 Contributions

This thesis makes a number of contributions in the field of 3D morphable model fitting:

We present an end-to-end pipeline for real-time 3D face reconstruction from single images as well as from monocular in-the-wild videos, building upon the linear shape identity fitting of Aldrian & Smith [AS13]. We develop a real-time 3D shape-from-landmarks fitting algorithm that explicitly tackles scenarios occurring when fitting to in-the-wild images, first and foremost facial expressions and 2D contour correspondences. To that end, we propose an iterative strategy to fit expression blendshapes and identity parameters, as well as a dynamic approach to facial contour landmarks fitting. With the lowest mesh resolution and a minimum number of iterations, the proposed algorithm has a run time of below 1 millisecond on a regular desktop CPU, making it possible to fit the model at 1000 frames per second (fps). Using a higher resolution model, and a recommended amount of 5 iterations of the algorithm, it still achieves 80 fps, which is orders of magnitude faster than current existing algorithms. We show experimentally the convergence of the iterative algorithm, and the effectiveness and importance of the proposed expression and contour fitting. The overall algorithm exhibits 3D shape reconstruction accuracy on par or exceeding that of algorithms that run orders of magnitudes slower and use learning-based or nonlinear optimisation methods.

We then extended the proposed fitting to multiple images, using the knowledge that there is one identity to reconstruct amongst all frames. The linear formulation of the shape identity fitting is extended so that each

frame contributes to the solution, while solving for one set of PCA identity coefficients. We demonstrate that, using the proposed method, one is able to achieve state of the art performance, and we show that recovering the shape from landmarks is on par or of superior accuracy to that achievable by existing more complex, often nonlinear algorithms that optimise for appearance error, and are often much slower. Further, to reconstruct face appearance, a simple weighted-mean based approach is presented to fuse textures from multiple images, which is applicable in a real-time context.

Finally, the research in this thesis is published as a lightweight open source library for 3D morphable face model fitting, being one of very few public and maintained 3DMM fitting frameworks and thus constituting a significant contribution to the research community.

Together, these innovations enable a range of new applications in consumer applications, augmented and virtual reality, or human-computer interaction, where real-time interaction and processing are of paramount importance.

1.2 List of Publications

The following publications have directly emerged from this thesis:

- *Real-time 3D Face Fitting and Texture Fusion on In-the-wild Videos*, P. Huber, P. Kopp, W. Christmas, M. Rätsch, J. Kittler, IEEE Signal Processing Letters, 2017 [HKC+17]

- *Real-time 3D Face Super-resolution From Monocular In-the-wild Videos*, P. Huber, W. Christmas, M. Rätsch, A. Hilton, J. Kittler, ACM SIGGRAPH 2016 Posters, 2016, Anaheim, United States [HCH+16]

- *A Multiresolution 3D Morphable Face Model and Fitting Framework*, P. Huber, G. Hu, R. Tena, P. Mortazavian, W. Koppen, W. Christmas,

M. Rätsch, J. Kittler, International Conference on Computer Vision Theory and Applications (VISAPP) 2016, Rome, Italy [HHT+16]

- *Fitting 3D Morphable Models using Local Features*, P. Huber, Z. Feng, W. Christmas, J. Kittler, M. Rätsch, IEEE International Conference on Image Processing (ICIP) 2015, Québec City, Canada (Recognised as a 'Top 10%' paper) [HFC+15].

Furthermore, the following publications resulted from collaboration with other researchers in the field:

- *Dynamic Attention-controlled Cascaded Shape Regression Exploiting Training Data Augmentation and Fuzzy-set Sample Weighting*, Z. Feng, J. Kittler, W. Christmas, P. Huber, X.J. Wu, IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2017, Honolulu, USA [FKC+17]

- *Efficient 3D Morphable Face Model Fitting*, G. Hu, F. Yan, J. Kittler, W. Christmas, C. Chan, Z. Feng, P. Huber, Elsevier Pattern Recognition, 2017 [HYK+17]

- *3D Morphable Face Models and Their Applications*, J. Kittler, P. Huber, Z. Feng, G. Hu, W. Christmas, International Conference on Articulated Motion and Deformable Objects (AMDO) 2016, Palma de Mallorca, Spain [KHF+16]

- *Random Cascaded-Regression Copse for Robust Facial Landmark Detection*, Z. Feng, P. Huber, J. Kittler, W. Christmas, X.J. Wu, IEEE Signal Processing Letters, Vol:22(1), 2015, pp. 76-80 [FHK+15]

- *Report on the FG 2015 Video Person Recognition Evaluation*, J. Beveridge et al., IEEE International Conference on Automatic Face and Gesture Recognition (FG) 2016, Ljubljana, Slovenia [BZD+15].

1.3 The *eos* Open-source Fitting Library

3D morphable face models and software to perform 3D face model fitting are both not readily and openly available. This is in stark contrast to 2D methods, like ASMs or AAMs, for which numerous open source models and implementations exist. Partly, this is due to the fact that generating a 3D face model is much more involved than generating a 2D model, and 3D model fitting algorithms are much more complex than their 2D counterparts.

During this thesis, a lightweight open-source software library for 3D model fitting was developed[1]. It is amongst very few available and maintained 3D face model software packages. The library is a lightweight header-only 3D Morphable Face Model fitting library, written in modern C++11/14. Apart from its native C++ interface, it contains Python and Matlab bindings and is thus usable from most programming languages prevalent in the computer vision community. The library contains functionality to use 3D morphable face models, and the shape-to-landmarks fitting algorithm presented in this thesis, offering an open-source solution to real-time 3D morphable shape model fitting.

[1] The library is called *eos*, and it is available on GitHub under Apache License 2.0: `github.com/patrikhuber/eos`

Together with the library, we published the *Surrey Face Model* (SFM), and a low-resolution shape-only version of the SFM is available as part of the open source repository, free for non-commercial use. The low-resolution morphable shape model, consisting of 3448 vertices, is particularly useful for real-time applications. Additionally, we provide a set of six expression blendshapes, which enable the use of the model for in-the-wild imagery.

1.4 Outline

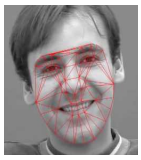
This thesis is organised as follows. The following chapter, Chapter 2, gives an overview of the state of the art in 3D face reconstruction from single images and from monocular videos, focusing on 3D morphable face models.

Chapter 3 contains a brief introduction to 3D morphable face models, and their parts that are used throughout the thesis. Chapter 4 then presents our novel approach to real-time 3D morphable face model fitting, focusing on reconstruction from single photographs. In Chapter 5, we extend the approach to multiple images and videos by considering a closed-form solution to shape fitting incorporating all images. The chapter also introduces a method for texture fusion from multiple images, incorporating pose and quality of frames, to perform multi-view fitting in real-time. Chapter 6 summarises and concludes the findings of the thesis. In Chapter 7, we outline drawbacks of the proposed approach and present possible future research directions.

Chapter 2

Related Work

Face modelling, analysis and reconstruction have been researched for nearly three decades. In 1991 and 1992 respectively, the landmark papers of Eigenfaces [TP91] and Active Shape Models [CT92, CTCG95] were to define the research landscape for years to come. In their seminal work on Eigenfaces, Turk & Pentland proposed to apply PCA to a set of face photographs and thus learn a 2D statistical model of face variations, and used their algorithm for face recognition, face completion, and face tracking. Cootes & Tailor, in their inaugural publication about Active Shape Models (ASM), proposed to model the statistical distribution of points on the face with a shape model, and derived an algorithm to fit this learned model to novel face images. Subsequent work extended this to Active Appearance Models [CET98, CET01], which learns a statistical model of face texture alongside the shape model.



Landmarks and shape of a 2D Active Appearance Model.

In 1999, Blanz & Vetter [BV99] published their seminal work on 3D Morphable Face Models, which builds a statistical model of 3D face shape and colour from densely registered 3D scans, and presents an algorithm to reconstruct novel faces from a single 2D input photograph. Since then, numerous methods have been proposed for modelling and analysing faces and for reconstructing them from 2D input data. Particularly for reconstruction from a single 2D input photograph, most approaches use a face model or

face template, since the task is to recover 3D face shape information lost through projection. This chapter thus mainly focuses on model-based, and specifically 3D Morphable Face Model based face fitting. Some of the most related non-model based reconstruction approaches are briefly touched upon as well.

2.1 Single Image 3D Morphable Face Model Fitting

Reconstructing 3D information from a single 2D photograph is an inherently ill-posed problem, requiring some form of prior knowledge to make the task feasible. For this reason, many methods use a 3D face template or a statistical face model of some sort to reconstruct a face in 3D. This section covers mainly methods that use a 3D Morphable Face Model for this task, which are most relevant to this thesis. In this setting, the process of recovering 3D pose and model parameters from an input image is often referred to as *model fitting*, or simply *fitting*.

In their inaugural paper introducing the 3D Morphable Model, Blanz & Vetter [BV99] proposed to use a non-linear cost-function optimising for shape, albedo, perspective camera, Phong illumination, and a colour transformation to fit the model to a 2D image. In an analysis-by-synthesis way, the cost function aims to minimise the RGB colour differences between the projected model vertices, given the current parameter estimates, and the input image. They optimise this cost function with stochastic gradient descent and a coarse-to-fine strategy with respect to model resolution, number of coefficients used for fitting, and prior weights. Also, in the last iterations, the face model is broken down into segments and the coefficients independently optimised. The fitting is initialised by a rough manual alignment of the average 3D head.

The subsequent landmark in fitting algorithms is the 2005 proposed

method by Romdhani & Vetter [RV05], sometimes called *Multiple features fitting*. In addition to using pixel intensities, various image features such as edges or the location of specular highlights are used to construct a cost surface that is more smooth and with fewer local minima. The fitting is initialised with manually clicked landmarks, and the cost function is minimised using a Levenberg-Marquardt optimisation algorithm. In [Kno09], Knothe follows a similar approach, but proposes a segmented model, termed Global-to-Local (G2L), to increase the representation and generalisation capacities of the model. An adaptive fitting method is presented that estimates the model parameters using a multi-resolution approach, increasing accuracy and efficiency, with a run time of around 30 seconds per image.

In 2013, Aldrian & Smith [AS13] proposed to decompose the fitting process into geometric and photometric parts, and state the problem as multilinear system that can be solved accurately and efficiently. They use an affine projection to model the camera, and fit the shape only to landmarks, which has a closed-form solution. Illumination is modelled with spherical harmonic basis functions, with which illumination and albedo can be estimated in closed-form. Hu et al. [HYK+17] similarly decompose the fitting into mostly linear sub-parts, while using a perspective projection camera model.

Recently, Bas et al. [BSBW16] extended the approach of Aldrian & Smith by using edges to obtain a more precise shape. The shape is fitted to landmarks as well as image edges from an edge detector, which improves the shape fitting wherever image contours are present, notably along the boundaries of the face. In that work, they estimate only shape, and refrain from estimating albedo or illumination, and use a scaled orthographic projection instead of the previously used affine projection.

A different approach that has been explored in the past few years is

using Markov chain Monte Carlo (MCMC) methods to solve the highly complex fitting problem. Schönborn et al. [SFEV13, SEMFV16] use MCMC-sampling to fit pose, shape, albedo and illumination, whereby proposals of new parameter values are generated that are accepted or rejected according to a Metropolis-Hastings criterion. Their approach is able to directly incorporate uncertain face detection and landmark detection results (for example, multiple face bounding boxes). Usually, a few thousand samples need to be drawn to approximate the probability distribution of the parameters, and the resulting run time is in the order of several minutes per image. They use their approach for face recognition, face attributes estimation (for example age or hairstyle) and gaze estimation [ESFV14]. In subsequent works, they incorporated an explicit background model into their fitting ([SEFV15]), and means to handle occlusions ([ESB+16]).

Recently, learning-based methods have been explored for 3DMM fitting. For example, Huber et al. [HFC+15] and Zhu et al. [ZYY+15] learn regressors that regress from image features (like HOG, Histogram of Oriented Gradients) to 3DMM parameter updates. Deep-learning based approaches have also gained popularity amongst 3DMM fitting methods. For example, in their follow-up work, *3D Dense Face Alignment (3DDFA)*, Zhu et al. [ZLL+16] employed a convolutional neural network to regress shape model and camera parameter updates. While the last work shows promising results, in general, these methods suffer from the lack of availability of ground truth training data that mirrors the conditions that are present when employing the fitting on in-the-wild data. These learning-based methods need ground truth of 3D model parameters (or ground truth 3D meshes), which only exist in the form of databases consisting of 3D scans captured in lab conditions, and thus, the methods do not generalise well to in-the-wild test images.

Most recently, Booth et al. [BAP+17a, BAP+17b] tackled the problem of learning a 3DMM texture model that is suitable for fitting to in-the-wild images. Their texture model is learned from in-the-wild images, and they do not employ an illumination model, thus avoiding the discrepancies caused by using an albedo model captured from 3D scans acquired in lab conditions and then applying such a model, together with an illumination model, to in-the-wild images. Their approach also comprises of a promising Gauss-Newton optimisation strategy, the Project-Out method, to solve the resulting nonlinear optimisation problem.

2.2 Multi-frame Reconstruction

Reconstructing 3D from 2D data becomes less ill-posed when information from multiple images is available, in particular if these images are from distinct view points. Structure from Motion methods, like for example Bundle Adjustment, are able to recover static 3D scene geometry from multiple 2D viewpoints, using no prior knowledge about object geometry (e.g. [TMHF99, SCD+06, SSS08]). A similar, but more difficult case, is when the object to be reconstructed is not rigid, for example a human body or animal, and changes its shape over multiple images. These kinds of problems are typically tackled with nonrigid Structure from Motion algorithms (e.g. [BHB00, Bra01, THB03]).

The task can be made slightly easier by injecting prior knowledge that the object to be reconstructed is a human face, and 3DMMs are an excellent tool for this purpose. They can model the general face shape as well as shape variety amongst different individuals, and, optionally, facial expressions, accounting for the non-rigidity. While Blanz & Vetter [BV99] mention fitting the model to multiple images, it is not further explored in their original work. Amberg [Amb11] is among the first who research 3DMM

fitting to videos. They construct a nonlinear cost function which solves for a single set of identity and albedo parameters for a given video, and solve it by optimising over 30 frames of a video at a time. Their work shows very promising results in somewhat unconstrained, but limited, scenarios, and requires landmarking of a minimal set of frames manually. Van Rootseler et al. [vRSV11, vRSV12] research 3DMM fitting to multiple images in the context of face recognition. They explore two options of extending the traditional analysis-by-synthesis fitting approach to multiple images: *a)* by averaging the fitted PCA shape and colour coefficients from two images, and *b)* by extending the traditional non-linear fitting cost function to include two images, optimising for only one set of PCA shape and colour coefficients. In a very small-scale evaluation on FRGC and MultiPIE, they were able to demonstrate benefits using both approaches. Herold et al. [HDG+12, HDG+14] follow a different approach, and investigate the use of particle filters to fit a 3DMM under different views and time. They perform an evaluation on synthetic data, but no follow-up work on real data exists.

In an a bit different, but pioneering work, originating from the computer graphics and animation community, Garrido et al. [GVWT13] present an approach to reconstruct 3D face geometry using personalised blendshape models. By additionally applying temporally coherent optical flow and photometric stereo, they are able to reconstruct fine face details such as wrinkles and laugh lines, at the cost of a run time of around 10 minutes per frame. Their approach requires significant manual, subject-specific training and labelling, and results are shown in controlled, frontal scenarios, from high quality cameras. Ichim et al. [IBP15] present a similarly groundbreaking work for high-detail recovery of 3D face avatars from hand-held videos. Their optimisation integrates feature tracking, optical flow, and

shape from shading, over all frames of a video. Fine-scale details such as wrinkles are captured separately in normal maps and ambient occlusion maps. They present results under very controlled conditions, and mostly frontal poses. Their approach requires manual interaction of about 15 minutes per video, and the run-time is in the order of an hour for a short video. In a similar work, Wu et al. [WBGB16] construct a local model with an anatomical constraint to obtain more flexibility and expressiveness than global models can offer. Their method requires about 1-2 minutes to reconstruct one frame with a very high resolution mesh (around 700,000 vertices).

Last, in the approach of Booth et al. [BAP+17a, BAP+17b], the authors also present results on a video dataset by applying the fitting on a per-frame basis.

2.3 Real-time Methods

Real-time 3D face reconstruction from monocular video streams was not feasible up until a few years ago. Initial advances have been made by Xiao et al. [XBMK04] and Matthews et al. [MXB07], which present their approach of ”combined 2D+3D AAM“ fitting. They propose a real-time algorithm to fit 2D AAMs while constraining them with 3D shape modes of a rudimentary 3DMM. The fitting is computed on a per-frame basis and their algorithm achieves around 60 fps. Their work also provides an excellent overview of the trade-offs between 2D and 3D models. While the results are promising, their evaluation is mostly done on images with homogeneous background in relatively constrained conditions.

Significant progress was made with the broad availability of low cost depth sensors, or RGB-D cameras, most notably the Microsoft Kinect. For example, Weise et al. [WBLP11] present one of the first works on real-time,

markerless 3D shape reconstruction using RGB-D sensors, aimed at the task of facial reanimation. In a subsequent work, Bouaziz et al. [BWP13] remove the necessity of the previous approach to learn a user-specific model in advance by using a dynamic, adaptive expression model, together with an identity PCA model. Other follow-up work exists, however, this thesis focuses on the relevant research which works on monocular video, without requiring a depth sensor.

Major contributions were made by Cao et al. from 2013 onwards. In one of their first works, [CWLZ13], they present an approach to 3D shape regression for real-time facial animation. In a tedious preprocessing step, user-specific training images and blendshapes had to be captured, which are then applied at run-time, together with a generic 3D shape regressor. In their follow-up work, [CHZ14], they remove the need for the preprocessing step by introducing a two-step approach of alternating 3D shape regression and dynamic adaptation of user-specific blendshapes. Their approach is shown to be able to deal with notable variations in pose and illumination conditions, as well as occlusions. However, no evaluation on any public in-the-wild database is done. Building up on this work, in [CBZB15], they then present an approach to model more fine-grained face details like wrinkles by introducing a local model on top of the previously presented more global method. Identical to the previous approach, it does not require any user-specific training, but the improved shape detail level comes at the cost of requiring CUDA and a powerful NVIDIA GTX 980 GPU to run in real-time. Additionally, results are only shown in frontal, highly controlled conditions.

In 2015, Jeni et al. [JCK15] present a purely 3D-based regression method, with consistent landmarks across all poses. Their promising approach combines dense cascade-regression-based face alignment with a 3D part-

based deformable model fitting. However, they use rendered 3D meshes to train their algorithm, which do not contain the variations that occur in 2D in-the-wild images; for example, the meshes have to be rendered on random backgrounds. Thus, their algorithm has only been evaluated in somewhat constrained conditions.

The most recent work of real-time 3DMM fitting is Thies et al. [TZS+16]. They present an approach to 3D face reconstruction and face re-enactment, where they track the subject in a source video in real-time and re-enact a different subject in the target video with the facial expressions from the source actor. In the source video, they employ an impressive real-time analysis-by-synthesis 3D morphable face model fitting, which minimises the RGB image error over the face pixels — essentially performing the traditional 3DMM fitting, which normally takes in the order of a minute per image, in real-time. The cost function consists of the mentioned photo-consistency term, a landmark constraint, and regularisation priors on the PCA shape, albedo and expression coefficients. According to the authors, the first few frames of a video are used as initialisation period, where they fit the albedo and shape model, which is then kept constant during the remainder of the tracking, simplifying the problem. The real-time efficiency is achieved with a highly data-parallel GPU-based Iteratively Reweighted Least Squares solver, which they implement as a highly customised solution on an NVIDIA Titan X graphics card. Their approach, run with a commercial 2D facial landmark tracker, demonstrates impressive performance in their limited test setup.

2.4 Available 3D Morphable Face Models

Since this thesis entails a 3D Morphable Face Model and a publicly available fitting algorithm, it is worth giving a brief overview of the state of the art

of published models. While there exists a number of other techniques to model faces, we focus on 3DMMs or approaches very close to them.

In 2009, Paysan et al. published the perhaps most well-known Basel Face Model (BFM) [PKA+09]. It consists of a 3DMM with shape and albedo PCA models built from 200 example scans, and the mesh contains the face, ears and neck area, totalling 53490 vertices. The model is widely used in the community and available on request for research institutions. It does not contain the ability to model facial expressions, and does not come with any fitting algorithm.

From 2014 to 2016, Wuhrer et al. released various statistical 3D shape models [BSBW14, BW15, BW16]. All of their models are trained on the BU-3DFE face database [YWS+06], and most models also contain scans from the Bosphorus database [SAD+08]. Their work focuses on 3D registration, and not on 3D to 2D fitting. Their licence allows use of their models for non-commerical research purposes.

At the same time, Cao et al. [CWZ+14] published FaceWarehouse, a dataset of 3D face scans, including expression scans, captured from a Kinect camera. While they do not make their learned model available, they provide registered meshes as part of their database, so a model can easily be learned from them. Scans from the FaceWarehouse database are often used to add an expression model to the BFM.

Most recently, Booth et al. created 3D morphable models learned from 10,000 faces [BRZ+16]. With the help of their massive amount of data, they created a global model as well as bespoke models tailored by age, gender and ethnicity. Their models are available on request to researchers involved in medically oriented research.

With regards to model fitting algorithms, very little software exists.

The software of Muré[1] is a sometimes mentioned resource of a 3DMM fitting algorithm. However, in their report ([Mur12]), it is stated that the implementation is not working correctly. This is only an indication of the complex task of implementing a non-linear 3DMM fitting algorithm in the likes of Blanz & Vetter [BV99]. The perhaps most promising resource of a public 3DMM fitting algorithm is the very recently published MATLAB code of Bas et al. [BSBW16]. They provide an open-source implementation to fit the shape model of the BFM to any photographs, given 2D facial landmarks as input. Their algorithm takes of the order of a minute per image, and it does not handle facial expressions.

2.5 Summary

Single-image 3D morphable face model fitting from in-the-wild images presents an ongoing challenge. Existing analysis-by-synthesis based approaches have only demonstrated very limited success on in-the-wild imagery. We identify a number of key problems:

Albedo and illumination model The traditional albedo model of 3DMMs is built from 3D scans captured in lab-conditions. There is a large discrepancy between the appearance variations captured by these scans, and the conditions present in in-the-wild images. Furthermore, these scans are usually captured with no or little facial hair and with no hair occluding the face area. Last, it is unclear which illumination models (for example the Phong model or Spherical harmonics) are adequate for in-the-wild images.

Cost function & optimisation The task at hand is a complex nonlinear optimisation problem, with many parameters involved. A plethora of

[1] `https://github.com/MichaelMure/3DMM`

different cost functions have been devised, incorporating multi-resolution approaches and various terms like image edges or silhouette constraints — all with the goal of making the optimisation more robust and converge to a better optimum. The importance of the choice of cost function and optimiser, particularly in conjunction with the mentioned problems with the albedo model, has not yet been sufficiently analysed.

Imaging conditions The occurrence of external occlusions and glasses make the analysis-by-synthesis approach problematic on in-the-wild images. Additionally, strong landmark constraints or an explicit background model (see [SEFV15]) are needed to prevent shrinking of the model during fitting.

The MCMC-based fitting approach of Schönborn et al. tackles many of these issues, but at the cost of a run time even slower than the traditional methods. The approaches that decouple the fitting into linear sub-steps show promise, but at the cost of decoupling the shape reconstruction from appearance completely, recovering the shape only from landmarks. Additionally, many of the state of the art works in 3DMM fitting do not tackle problems occurring in in-the-wild images like strong facial expressions or the mismatch of vertex correspondences from the facial contour detected by a 2D landmark finder and 3D mesh vertices.

Learning based methods have shown promise to avoid the issues of the nonlinear optimisation task. However, the new issue arises that these methods require 3D ground truth to be trained on, and that 3D ground truth is not available for in-the-wild imagery. Hence, these methods have only demonstrated limited applicability under difficult conditions so far.

With regards to multi-image 3D face reconstruction, most algorithms originating from the graphics community are able to produce high-detailed 3D shape output, but require in the order of minutes and hours to process

a video, and are thus targeted at different applications. Additionally, many require tedious manual interaction, and are meant for more controlled scenarios. There has been work done on multi-image 3DMM fitting, but in general, it has not yet been evaluated systematically.

Significant advances have been made in real-time 3D face tracking and reconstruction over the last few years. However many of the approaches suffer from the same issues as the single-image methods — most approaches are not evaluated on true in-the-wild imagery, require 3D ground truth for training, a powerful GPU to run in real-time, or are heavily dependent on good, often commercial, 2D facial landmark detections.

It is also noteworthy that 3D face models have by far not been adopted as widely as their 2D counterparts. Not many research groups possess the manpower required to construct a 3DMM, and most of the few existing 3DMMs are only available after signing licence agreements. Even after obtaining a 3DMM, there exists only the open-source algorithm of Bas et al. to perform model fitting (which was released one year after ours), but no code for real-time fitting or a framework to interact with 3DMMs is openly available. We believe these considerable hurdles are significantly impeding the more widespread adoption of 3D methods and 3D morphable face models specifically and this thesis aims to tackle some of these issues.

Chapter 3

3D Morphable Face Models

A 3D Morphable Face Model is a statistical model of face shape and appearance, built from a set of registered 3D face scans. PCA is applied to both the shape and vertex-colour data to obtain a model which spans a low-dimensional linear subspace of face shape and appearance, learned from example 3D scans. One can *morph* between these faces in PCA space, transfer face characteristics from one face to a different face, or generate new faces, which gave the model its name, *morphable* model.

In this chapter, we will briefly introduce the most important steps of model construction and define the components of a Morphable Model used throughout this thesis.

3.1 3D Mesh Registration

A 3D Morphable Model is typically built from a number of 3D face scans, usually captured in a lab with a stationary high-resolution 3D face scanner. For the Surrey Face Model, used throughout this thesis, the 3D scans have been acquired using a 3dMDface[1] camera system that consists of two structured light projectors, 4 infrared cameras that capture the light pattern and are used to reconstruct the 3D shape, and two RGB cameras

[1] `http://www.3dmd.com/`. In particular, a 3dMDface system from around 2007 was used to capture the scans.

recording a high-resolution face texture. Half the cameras record the face from the left side, the other half from the right side, resulting in an almost 180° view of the face. The images are acquired under somewhat uniform illumination to ensure that the model texture is representative of face skin albedo only. The 3dMDface software reconstructs a textured 3D mesh from this information.

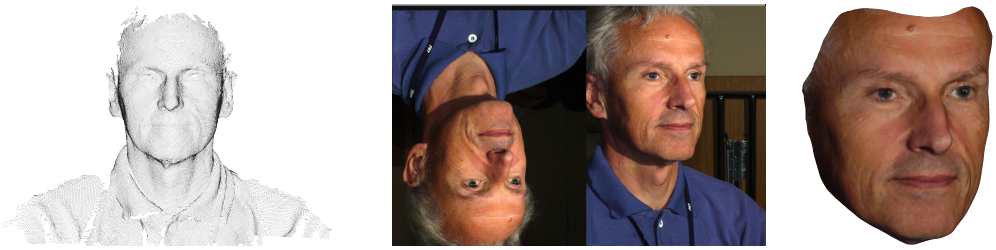


Figure 3.1: *(left)*: Raw mesh output from the 3dMDface software. *(middle)*: Texture image from two angles captured by the 3dMDface cameras. *(right)*: The scan and texture densely registered to a 3D reference mesh.

After these high-resolution 3D scans have been acquired, a key property of a 3DMM is that the scans have to be brought into dense correspondence before building the face model. A 3D to 3D shape registration algorithm is used for this purpose, in our case the Iterative Multi-resolution Dense 3D Registration (IMDR) method from Tena et al. [THHI06]. Figure 3.1 shows an example scan with the captured mesh on the left, the RGB texture in the middle, and the scan after registration to the 3D reference mesh. As this thesis makes use of the model and its multiple resolution levels, the mesh registration algorithm is in the following briefly outlined.

The IMDR algorithm uses a deformable reference 3D face model and performs a combination of global mapping, local matching and energy-minimisation to establish dense correspondence among all the scans at different resolution levels. The generic reference face we used consists of 845 vertices and 1610 triangles. The following is a high-level overview of the process:

1. The target scan is denoised using Gaussian and median filtering if spikes and noise are present.

2. A global mapping is performed from the generic face template to the target scan using facial landmarks, smoothly deforming the template mesh.

3. A local matching is done on the current resolution level based on the distances between reference and target vertices. If a particular vertex cannot be matched, its mirrored counterpart is used (and if that fails as well, the algorithm interpolates using the neighbouring matches).

4. The final set of matches guides an energy minimisation process that conforms the reference mesh to the target scan. Steps 3 and 4 are iterated.

5. The generic reference mesh is subdivided using a 4-8 mesh subdivision algorithm [VZ01].

6. Steps 3 to 5 are repeated until the desired highest mesh resolution is achieved.

This process results in a number of intermediary registration results, with the mesh resolution levels (number of vertices) 1724, 3448, 16759 and 29587. As part of this thesis, an even lower resolution model of 845 vertices has been generated, which corresponds to the number of vertices of the reference mesh. Figure 3.2 depicts three of these mesh resolutions with a close-up on the model's mesh, namely the meshes with 845 and 3448 vertices, and the highest resolution mesh with 29587 vertices. It is noteworthy that the higher resolution meshes are built upon the lower resolutions, and therefore each vertex from a lower resolution mesh is also present in all higher resolution meshes, and has the same vertex index.

For the Surrey Face Model, 168 captured and registered scans from earlier works were used. The recorded subjects represent a diverse range of skin tones and face shapes to well represent the multicultural make up of many modern societies. Non-Caucasian people are well-represented and significant numbers of subjects from other races are included allowing the model to generalise well to people from various backgrounds. This is one key difference to other existing 3DMMs, for example the Basel Face Model, which consists of a majority of Caucasian subjects.

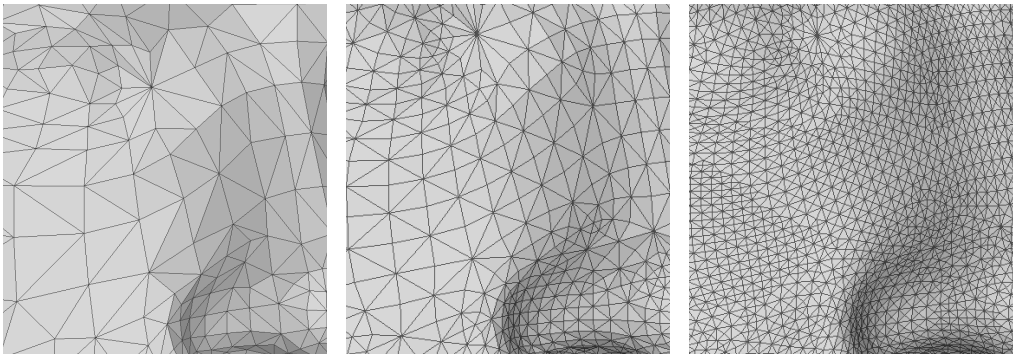


Figure 3.2: Close-up of the different mesh resolutions of the Surrey Face Model. *(left)*: The lowest-resolution model with 845 vertices. *(middle)*: Medium-resolution model (3448 vertices) *(right)*: The full resolution model (29587 vertices).

This section also highlights one core difference between 3DMMs and 2D-based methods like AAMs. To build a 3DMM, high-quality 3D scans are used that have been captured in the lab and under controlled illumination conditions — ideally, under homogeneous diffuse-only illumination, so that the resulting texture reflects what is called face albedo. That face albedo, when being rendered and lit with an illumination model like Phong illumination or Spherical harmonics, makes up the final face appearance we are used to seeing. In practice, however, it is quite difficult to create such conditions, which is why the albedo model part of a 3DMM is also often simply referred to as *colour*- or *texture model*. To avoid ambiguity, in this thesis, we will refer to the PCA appearance model as *albedo* or *colour*

model, and use the word *texture* whenever RGB data from the original image is used (for example remapped onto the model). Furthermore, when capturing 3D scans of subjects, hair is usually tied back not to obstruct the face area, and often scans are taken without or with little facial hair present.

In contrast to these 3D models, 2D models are often built from in-the-wild images directly, which contain all the appearance variations that would be present in later test images as well. 2D models can be much more easily built since they only require 2D landmark annotations, which can easily and accurately be obtained. Illumination and face appearance are not separated, so that the resulting appearance model contains illumination variations and things like shadows too, which makes both model creation as well as model fitting a much easier task.

3.2 A PCA Model of Faces

After a 3D face scan has been brought into dense correspondence, its mesh is represented as a vector $\mathbf{S} \in \mathbb{R}^{3N}$, containing the x, y and z components of the shape, and a vector $\mathbf{T} \in \mathbb{R}^{3N}$, containing the per-vertex RGB colour information. N is the number of mesh vertices. All training meshes are then stacked in a data matrix, to which PCA is applied, separately to the shape and colour data. The resulting 3DMM consists of two PCA models, one for the shape and one for the colour information. Each PCA model

$$\mathcal{M} := (\bar{\mathbf{v}}, \boldsymbol{\sigma}, \mathbf{V}) \tag{3.1}$$

consists of the components $\bar{\mathbf{v}} \in \mathbb{R}^{3N}$, which is the mean of the example meshes, a set of principal components $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_m] \in \mathbb{R}^{3N \times m}$, and the standard deviations $\boldsymbol{\sigma} \in \mathbb{R}^m$. m is the number of principal components, so that 99% of the variance in the original data is retained. $m \leq n - 1$, where n is the number of scans used to build the model.

With this simple model, novel faces can be generated by calculating

$$\mathbf{S} = \bar{\mathbf{v}} + \sum_{i=1}^{m} \alpha_i \sigma_i \mathbf{v}_i \tag{3.2}$$

for the shape, where the vector $\boldsymbol{\alpha} \in \mathbb{R}^m$ conveys the *shape coefficients*, a set of 3D face instance coordinates in the shape PCA space. The same can be calculated for the colour model.

To analyse the variations in the built face model, we can visualise the directions of largest variance in the PCA space by taking the formula in Equation 3.2 and setting a specific α_i to a fixed value while setting all others to zero. The resulting face mesh $\mathbf{S}$ can then be rendered. Figure 3.3 shows the mean of the model and the first three shape components set to ± 2 standard deviations. The first components mainly account for global structure of the face, like global face shape (more round or square, slim or chubby) and size of the face. Later components model the finer structures of the face.

Figure 3.4 depicts the colour PCA model with the colour coefficients set to ± 2 standard deviations. Varying the first component of the colour model results mainly in a change of global skin colour from black to white, while the second component models more diverse changes relating to the gender. The third component encodes a mixture of skin colour and possibly gender.

For the reasons mentioned before, in this thesis, only the shape model of the 3DMM is used. For the face appearance, we use the RGB information directly extracted from images, as will be described in Section 3.4. The models mainly used throughout this thesis are the newly generated 845 vertices model, as well as the 3448 vertices model. Some figures use the 845 vertices model for the sake of a better visualisation.

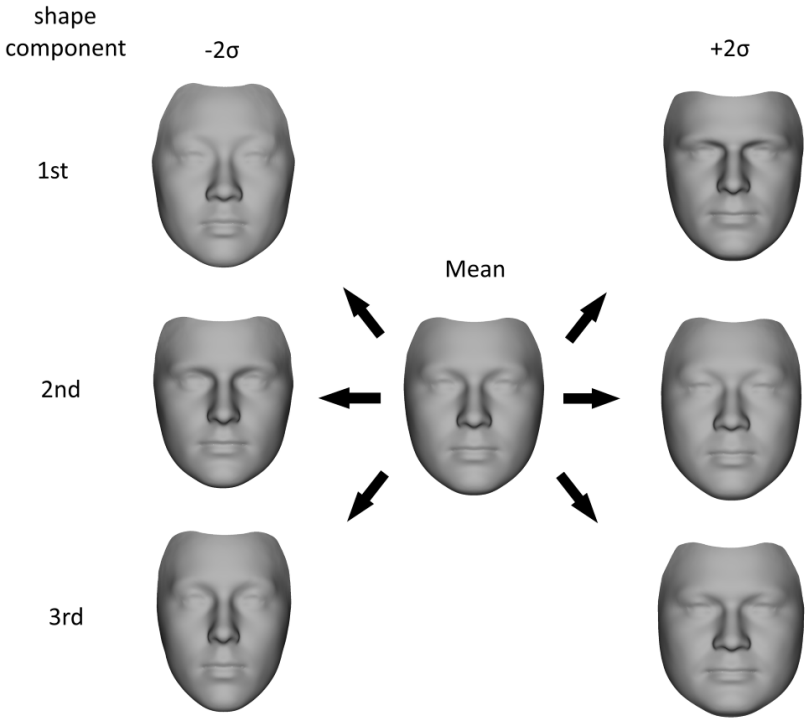


Figure 3.3: The mean face and shape variation of the high-resolution Surrey Face Model. The figure shows the first three PCA shape coefficients at -2 and +2 standard deviations.

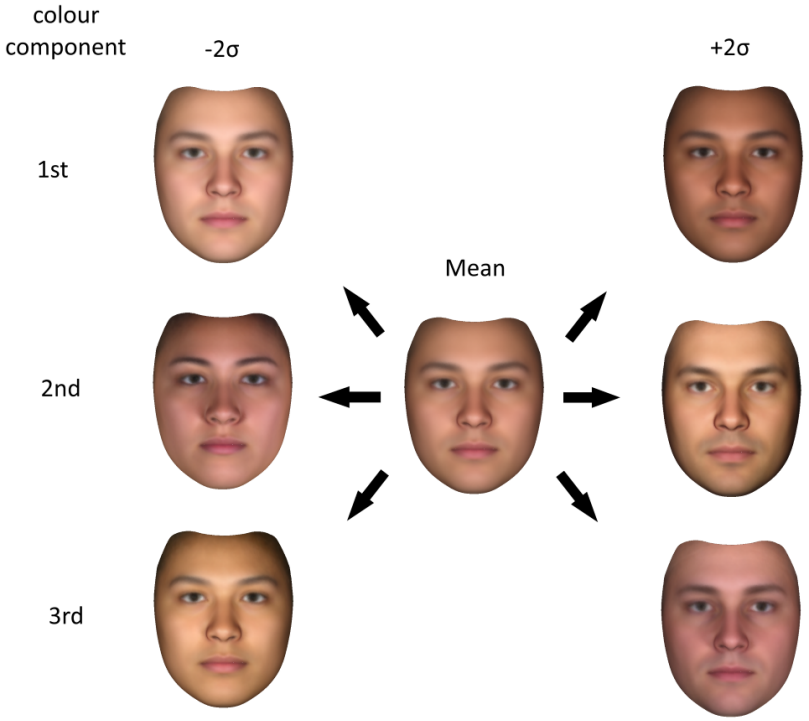


Figure 3.4: The mean face and colour variation of the high-resolution Surrey Face Model. The figure shows the first three PCA colour coefficients at -2 and +2 standard deviations.

3.3 Facial Expression Modeling

Many in-the-wild images contain subjects with non-neutral expressions, and in a video with a moving target, there will be a degree of facial expression visible in many frames, thus creating a need for the face model to be able to model facial expressions. Among others, there are two standard ways to handle expressions with a 3D Morphable Model, or face models in general:

A) Expression PCA: Expressions are modelled with an additional PCA model, learned from a set of 3D expression scans (e.g. [Rod07, BBPV03, AKV08]). This approach has the advantage that it takes into account that different subjects exhibit the same expression differently — e.g. two people's smiles are different. The expression PCA model forms a second PCA basis of expressions, separate from the PCA identity space described in the previous section. To fit expressions in a novel image, a set of PCA expression coefficients has to be estimated. One drawback of this approach is that the expression coefficients do not have a semantic meaning, i.e. a specific coefficient is not associated with a particular expression.

B) Blendshapes: Modelling expressions with linear expression blendshapes (see e.g. [PL06]), computed either from a set of 3D expression scans or crafted by hand. In this case, each expression is modelled with one blendshape, where a blendshape describes a linear offset in x, y and z for each vertex. During fitting, one blendshape coefficient has to be estimated for each expression blendshape, and the coefficients, $\boldsymbol{\psi} \in \mathbb{R}^k$, where k is the number of blendshapes, are equal or larger than zero, and usually we desire $\psi_j \in [0, 1]\ \forall j$. Blendshapes have the advantage that they can be precisely modelled by hand, if required, and the semantic information is not lost: one blendshape coefficient steers exactly one blendshape, and each blendshape has usually a direct semantic meaning (for example a "smile" expression, or a Facial Action Coding System blendshape (FACS,

[EF78, EFH02])). Additionally, new blendshapes can easily be added to a set of existing blendshapes, without having to re-learn the expression basis from all 3D scans.

The first approach is often employed in the domain of computer vision, while in computer graphics, there is a strong prevalence of blendshapes. In this thesis, we use linear expression blendshapes because of their simplicity and extensibility. We generate six expression blendshapes, representing the six *universal emotions* by Ekman [Ekm89, Ekm92]: anger, disgust, fear, happiness, sadness and surprise. These blendshapes are computed from 3D expression scans from the CVSSP 3D scan database [Rod07]. The database consists of 10 scans for each of the 6 expressions. To create the blendshapes, we compute the average amongst all 10 scans for each expression, resulting in 6 linear blendshapes (offset vectors). Because the vertices of the lower mesh resolution levels are subsets of the highest resolution levels, we can take the same subset of vertices on the expression blendshape vectors to obtain blendshapes for the lower model resolution levels. Figure 3.5 shows the six created expression blendshapes, added to the mean shape with each coefficient separately set to 1 while all others are kept at 0.

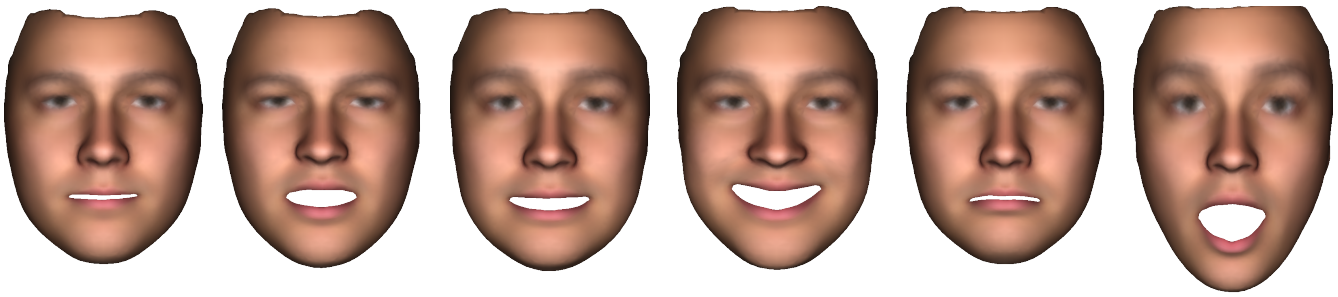


Figure 3.5: The six created expression blendshapes, visualised with the mean face. From left to right: anger, disgust, fear, happiness, sadness and surprise.

The blendshapes are stacked into a matrix $\mathbf{B}$, where each column represents one blendshape. A face shape $\mathbf{S}$ is then constructed by the PCA model (for identity) and a linear combination of the expression blendshapes:

$$\mathbf{S} = \bar{\mathbf{v}} + \sum_{i=1}^{m} \alpha_i \sigma_i \mathbf{v}_i + \sum_{j=1}^{k} \psi_j \mathbf{B}_j, \tag{3.3}$$

where $\mathbf{B}_j$ is the j-th column of $\mathbf{B}$ (the j-th blendshape) and ψ_j the corresponding blendshape coefficient.

3.4 Texturing

The PCA colour model is a useful representation for the appearance of a face, but in some cases it is desirable to use the pixel colour information (*texture*) from the image or a combination of the two. This may be particularly desired in light of the shortcomings of the PCA colour model discussed in Section 2.5. The texture from the input image remapped onto the mesh preserves all details of a face's appearance, while some high-frequency information is lost if a face is only represented using the PCA colour model. Another reason to use the texture is to avoid a colour and light model fitting, for example in consideration of run-time. Therefore, we would like a 2D representation of the whole face mesh that we can use to store the remapped texture. We create such a generic representation with the isomap algorithm of Tenenbaum et al. [TSL00]: it finds a projection from the 3D vertices to a 2D plane that preserves the geodesic distance between the mesh vertices. Our mapping is computed with the algorithm from Tena [Rod07].

In contrast to other representations, like for example cube mapping, this isomap has the advantage that it can be stored as a single 2D image, and it has face-like appearance, i.e. it can be easily used with existing face recognition and face analysis techniques. The isomap coordinates are only generated once, that is, the isomaps of different people are in dense correspondence with each other, meaning each location in the isomap corresponds to the same physical point in the face of every subject (for example, a hypothetical point $x = [100, 120]$ is always the center of the

right eye). This makes the isomap especially suitable for processing with further algorithms. Figure 3.6 shows the isomap of the 3448 vertices model as a wireframe.

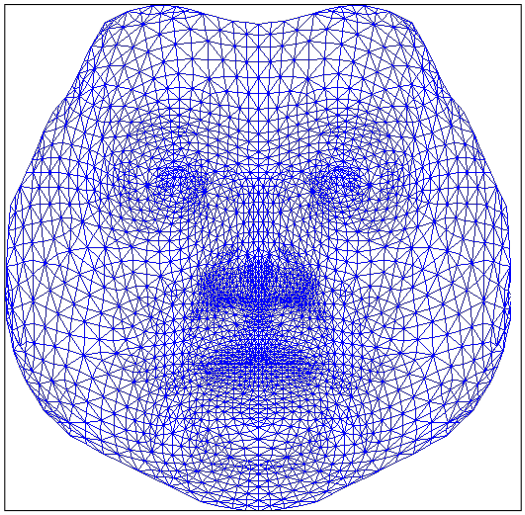


Figure 3.6: Texture representation in the form of an isomap. The 3D mesh vertices are projected to 2D with an algorithm that preserves the geodesic distance between vertices, resulting in a pose-independent, detail-preserving textural representation of a face. Shown is the isomap of the *sfm_3448*.

3.5 Summary

In this chapter, we introduced the 3D Morphable Model used in this thesis, and how we model facial expressions and texture. We created a low-resolution 3D shape model consisting of 845 vertices, which is best suited for real-time applications, when run time is of utmost importance. We further created 6 facial expression blendshapes, corresponding to Ekman's universal emotions, and we make these expression blendshapes, together with the 3448 vertices 3D shape model, publicly available as *Surrey Face Model*.

Chapter 4

Real-time 3D Shape Model Fitting

This chapter presents the proposed real-time shape-to-landmarks fitting algorithm. We introduce all the steps of our proposed algorithm: Pose estimation, shape identity fitting, expression fitting, and contour fitting. The proposed algorithm is then thoroughly evaluated, and we demonstrate state-of-the-art results with our shape-to-landmarks fitting compared to a more complex, non-linear fitting method.

In particular, we present two contributions most important to fitting to landmarks from in-the-wild images: A multi-linear solution to expression blendshapes fitting that alternates with shape identity fitting, and a dynamic method using 2D contour landmarks, often output by facial landmark detectors, in the 3DMM fitting. These 2D face contour landmarks are especially important to accurately reconstruct the global face shape (e.g. the face width). Overall, each of the steps contributes to the real-time capability of the proposed fitting method and to robust fitting results on a variety of in-the-wild images.

4.1 Orthographic Camera Model

To fit a 3DMM to a 2D image, an explicit camera or imaging model is needed, that projects the model from 3D space to 2D image space. The pose of a face is thus modelled explicitly and separate from object shape. When fitting the model to a 2D image, these camera or pose parameters have to be recovered together with the face shape parameters.

The camera model projects a face instance $\mathbf{S}$ from its model-centred 3D coordinates to coordinates in the image plane. It can be expressed with a function $\mathcal{P}(\mathbf{S}, \boldsymbol{\rho}) : \mathbb{R}^{3N} \rightarrow \mathbb{R}^{2N}$, where $\mathbf{S}$ is the shape instance and $\boldsymbol{\rho}$ a set of camera parameters. Various camera models can be chosen — for example, common choices are perspective, (scaled) orthographic, or affine projection. The projection of a 3D point or vertex $\mathbf{v} = [x, y, z]^{\mathrm{T}}$ from the shape instance $\mathbf{S}$ to a 2D point in the image, $\mathbf{v}' = [x', y']^{\mathrm{T}}$, can be expressed in two steps: First, with a rotation and translation into the camera coordinate system:

$$\mathbf{v}_c = [x_c, y_c, z_c]^{\mathrm{T}} = \mathbf{R}\mathbf{v} + \mathbf{t}, \tag{4.1}$$

where $\mathbf{R} \in \mathbb{R}^{3\times3}$ is a rotation matrix and $\mathbf{t} = [t_x, t_y, t_z]^{\mathrm{T}}$ is a translation vector.

Then, a camera projection is applied:

$$\mathbf{v}' = \pi(\boldsymbol{\rho}_{\mathrm{intr}}, \mathbf{v}_c), \tag{4.2}$$

where $\boldsymbol{\rho}_{\mathrm{intr}}$ is a set of intrinsic camera parameters that depend on the camera model that is used. In case of scaled orthographic projection, the intrinsic parameters consist only of a scale parameter s, and the projection becomes:

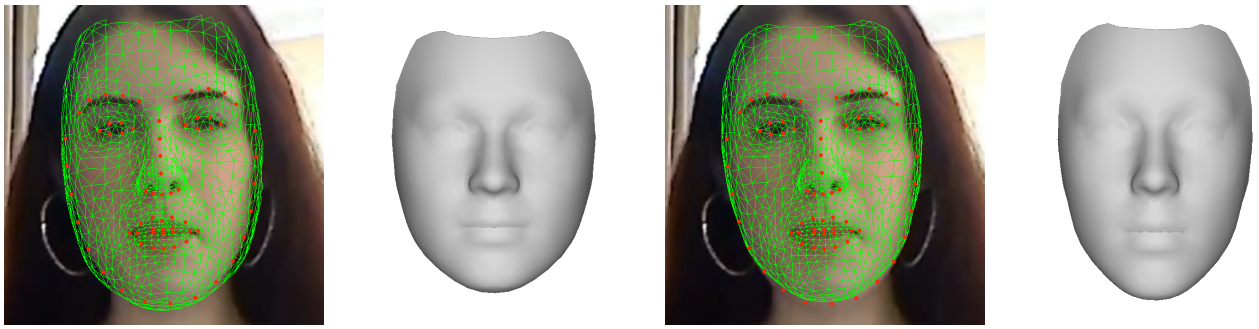
$$\mathbf{v}' = \pi(s, \mathbf{v}_c) = s \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \mathbf{v}_c. \tag{4.3}$$

The full set of camera parameters is then: $\boldsymbol{\rho} = [s, \mathbf{R}, t_x, t_y]$ (t_z is omitted in the orthographic projection). The rotation, here expressed as rotation

matrix $\mathbf{R}$, can be expressed in various ways — for example in Euler angles, or, often preferred, as quaternions (see also next section). These parameters $\boldsymbol{\rho}$ have to be recovered when fitting the model to a novel image.

4.1.1 Motivation

In the following, we will motivate the choice of the scaled orthographic projection for this thesis. We will start by investigating some of the popular choices. First, an affine camera model is an attractive choice. The algorithm of choice is the *Gold Standard Algorithm* of Hartley & Zisserman [HZ03], which provides a way to find the least squares approximation of an affine camera matrix given a number of 2D–3D point pairs. This method has been successfully employed for morphable model fitting by Aldrian & Smith [AS13]. It is attractive because the algorithm has a closed-form solution. However, there is a compelling argument against its use: It does not constrain the estimated camera transformation, and the resulting faces can exhibit considerable shear, as well as non-uniform scaling in the x and y axes. In practice, we often observe that this results in unnatural deformations of the face, and shape variations (like e.g. a slim face) being explained by the camera parameters, instead of shape variation. Figure 4.1 shows an example of this behaviour, with a relatively narrow face. In Figure 4.1a, on the left, we can see the mesh from the model fitting with the affine camera estimation, projected back to 2D. The mesh fits relatively well to the face in 2D. However, when looking at the mesh in 3D, on the right, we can see that the face shape was estimated to be too wide. The narrowness of the fitted face on the left stems from the parameters estimated by the camera projection: it estimated s_x, the scaling in the (horizontal) x axis, to be $s_x = 1.22$, and the scaling in the (vertical) y axis $s_y = 2.40$, resulting in a good fit after projection with these parameters, but an incorrectly recovered 3D face shape.



(a) Affine projection([HZ03]) (b) Scaled orthographic projection (Chapter 4.1.2)

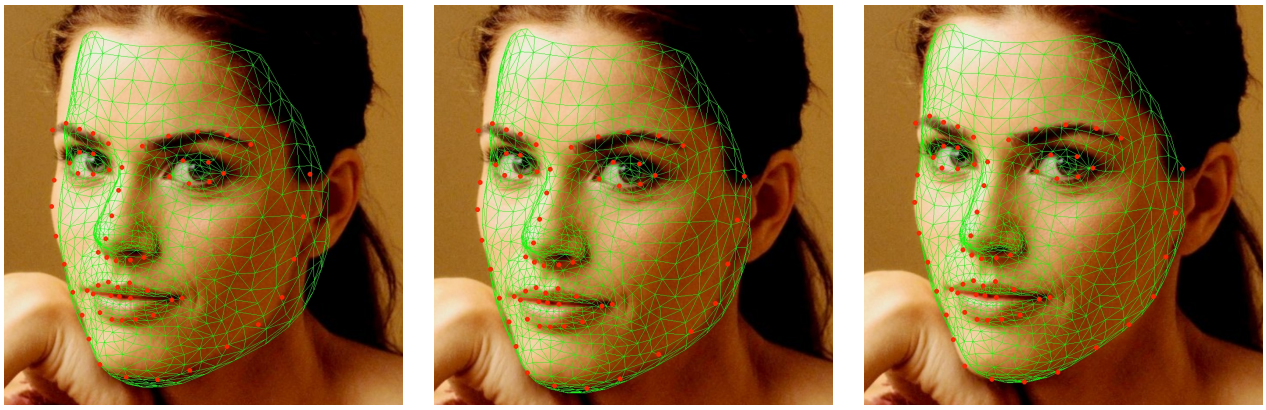
Figure 4.1: An issue of the affine pose estimation algorithm. The figure shows a relatively narrow face, fitted with the affine camera model and the scaled orthographic model. *(a)*: The affine projection includes non-rigid transformations like non-uniform scaling. This results in the narrow face being explained by the camera parameters, not the shape (i.e. the mesh on the right side of the subfigure is estimated too wide). *(b)*: The scaled orthographic projection only allows rigid camera transformations. The narrowness of the face is correctly explained with shape variation, as can be seen in the mesh visualisation on the right.

A popular alternative is to employ a scaled orthographic projection camera model. It is attractive because it constrains the space of possible transformations to the space of rigid motions in Euclidean space, $SE(3)$, and additionally allows for scaling. Figure 4.1b shows the same face picture, previously fitted with the affine camera model, now fitted with a scaled orthographic camera model. We observe that again the mesh fits well to the face in the 2D image. However, in this case, the actual 3D face shape is estimated much better. The narrowness of the face is explained by the shape identity parameters, and not by the camera projection.

When estimating the parameters of the scaled orthographic camera model, though the projection is linear, an orthogonality constraint has to be imposed on the rotation matrix $\mathbf{R}$ to limit the possible transformations to rigid rotations, i.e. the columns of $\mathbf{R}$ have to be orthogonal. In fact, using the 9 parameters of the 3×3 rotation matrix to specify rotation in 3D makes the problem overconstrained. A popular, but not necessarily good choice, is to parameterise the 3D rotation using Euler angles, specifically yaw, pitch and roll. However, Euler angles are not an ideal parameterisation for rotation.

The order in which they are applied needs to be clearly specified, and they are ambiguous: for example, a yaw rotation of 0° is the same as a rotation of 360° or 720°. Furthermore, they suffer from problems like gimbal lock, the loss of one degree of freedom. Often, another parameterisation is chosen for the rotation, namely unit or rotation quaternions. Rotation quaternions uniquely define a rotation in 3D with 4 parameters. They are commonly used in non-linear optimisation problems in computer vision (e.g. Bundle Adjustment, see Triggs et al. [TMHF99] for a more detailed discussion). Quaternions can easily be converted to rotation matrices or Euler angles, if required. Nonlinear solvers, like the Levenberg-Marquardt algorithm, are then used to estimate the camera parameters. However, successful convergence highly depends on the initialisation of the optimisation, the choice of the optimiser, its implementation, and a plethora of parameters that these nonlinear optimisers usually have. In practice, we have found that these optimisers often get stuck in local minima. Figure 4.2a shows an example where we estimate the scaled orthographic projection with the Levenberg-Marquardt solver from the Eigen library [GJ+10]. The rotation is parameterised with the three Euler angles. The algorithm fails to fully converge, and gets stuck in a local minimum. In Figure 4.2b, we show the result of employing the state-of-the-art nonlinear solver Ceres [AMO], with the rotation parameterised as unit quaternions. Ceres converges to a better solution, but the results still depend heavily on the choice of parameters and the input data. Figure 4.2c shows the result with the scaled orthographic projection estimation introduced in the next section.

Another popular choice is perspective projection. Perspective projection, in principle, suffers from the same problems as the orthographic projection, as a nonlinear optimiser has to be employed. In fact, the perspective projection introduces nonlinearity. Additionally, it introduces an additional



(a) Scaled orthographic projection (Eigen Levenberg-Marquardt)

(b) Scaled orthographic projection (Ceres)

(c) Scaled orthographic projection (Chapter 4.1.2)

Figure 4.2: Issues of different pose estimation algorithms. *(a,b)*: Non-linear optimisers often get stuck in local minima and are highly dependent on initialisation and choice of optimisation parameters. *(c)*: The two-step, iterative scaled orthographic projection estimation works well, and does not have any parameters to tune.

parameter, the focal length of the camera. Smith [Smi16] has shown that there is an ambiguity between the focal length, z-coordinates of a vertex, and the shape. For 3D shape reconstruction from a single image, it is therefore not a good choice, since it is a highly ambiguous problem, even with a 3D face model prior. In the work of Booth et al. [BAP+17a], a perspective camera model is used, but it is then explicitly mentioned that they found it beneficial to keep the focal length constant in most cases, due to its ambiguity with the z translation t_z of the camera.

In the following, we present our approach to solving for the scaled orthographic projection parameters, which we found to be much more robust, and successful on all images, without requiring any parameter tuning. Figure 4.2c shows an example of the pose fitting employed in this thesis. This approach has in parallel also been proposed by Bas et al. [BSBW16].

4.1.2 Closed-form Scaled Orthographic Pose Estimation

Our solution to estimating the scaled orthographic projection parameters consists of two steps. The problem is posed as the following: Given a set of 2D landmark locations and their known correspondences in the 3D Morphable Model, the goal is to estimate the pose of the face (or the position of the camera, which in our case is the same problem). To formulate the problem leading to a closed-form solution, we first operate under the assumption of an affine camera model. The *Gold Standard Algorithm* of Hartley & Zisserman [HZ03] provides a way to find a least squares approximation of a camera matrix given a number of 2D–3D point pairs.

We define the set of detected or labelled 2D landmark points in the image, $\{\mathbf{x}_i\}$, where each $\mathbf{x}_i \in \mathbb{R}^3$, and the corresponding set of 3D model points, $\{\mathbf{X}_i\}$, with each $\mathbf{X}_i \in \mathbb{R}^4$. Both are represented in homogeneous coordinates, i.e. with the last component set to unity. From these two sets of points, given $L \geq 4$ correspondences, we wish to determine the camera matrix $\mathbf{A} \in \mathbb{R}^{3\times4}$ that minimises $\sum_i \|\mathbf{x}_i - \mathbf{A}\mathbf{X}_i\|^2$, subject to the affine constraint $\mathbf{A}_3 = [0, 0, 0, 1]$, where $\mathbf{A}_3$ is the third row of $\mathbf{A}$. The two sets of points are first normalised by similarity transforms that translate the centroid of the image and model points to the origin and scale them so that the Root-Mean-Square distance from their origin is $\sqrt{2}$ for the landmarks and $\sqrt{3}$ for the model points respectively: $\tilde{\mathbf{x}}_i = \mathbf{T}\mathbf{x}_i$ with $\mathbf{T} \in \mathbb{R}^{3\times3}$, and $\tilde{\mathbf{X}}_i = \mathbf{U}\mathbf{X}_i$ with $\mathbf{U} \in \mathbb{R}^{4\times4}$. We then estimate a normalised camera matrix $\tilde{\mathbf{A}} \in \mathbb{R}^{3\times4}$. Each pair of correspondences $(\mathbf{X}_i, \mathbf{x}_i)$ contributes two equations:

$$\begin{bmatrix} \tilde{\mathbf{X}}_i^{\mathrm{T}} & \mathbf{0}^{\mathrm{T}} \\ \mathbf{0}^{\mathrm{T}} & \tilde{\mathbf{X}}_i^{\mathrm{T}} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{A}}_1^{\mathrm{T}} \\ \tilde{\mathbf{A}}_2^{\mathrm{T}} \end{bmatrix} = \begin{bmatrix} \tilde{x}_i \\ \tilde{y}_i \end{bmatrix}, \tag{4.4}$$

where $\tilde{\mathbf{A}}_i$ is the i-th row of $\tilde{\mathbf{A}}$, and $\tilde{x}_i$ and $\tilde{y}_i$ are the x and y coordinates of $\tilde{\mathbf{x}}_i$ respectively. Stacking these equations results in a system of equations

$\mathbf{A_8 p_8} = \mathbf{b}$, where $\mathbf{p_8} \in \mathbb{R}^8$ consists of the first two rows of $\tilde{\mathbf{A}}$. We compute the solution using the pseudo-inverse, yielding the first two rows of $\tilde{\mathbf{A}}$: $\mathbf{p_8} = \mathbf{A_8^+ b}$, and set the third row of $\tilde{\mathbf{A}}$ to $[0, 0, 0, 1]$. The final camera matrix is obtained by undoing the normalisation that was applied in the beginning: $\mathbf{A} = \mathbf{T}^{-1}\tilde{\mathbf{A}}\mathbf{U}$.

Computing the camera matrix $\mathbf{A}$ involves solving a linear system of equations — the algorithm calculates the least squares solution, so any number of corresponding points can be given. This process is also very fast, taking of the order of microseconds to compute. However, as described before, the estimated affine camera matrix $\mathbf{A}$ is not confined to the space of rigid motions in Euclidean space, $SE(3)$, which we would desire for our camera model.

In the second step, we thus constrain the pose estimation to rigid transformations, and in particular to a scaled orthographic projection. From the estimated affine camera matrix $\mathbf{A}$, following Bas et al. [BSBW16], we define the row vectors $\mathbf{r}_1 = \mathbf{A}_{1,1:3}$ and $\mathbf{r}_2 = \mathbf{A}_{2,1:3}$. We extract the scale parameter as $s = (\|\mathbf{r}_1\| + \|\mathbf{r}_2\|)/2$, and the translation as $\mathbf{t} = [\mathbf{A}_{1,4},\ \mathbf{A}_{2,4}]^{\mathrm{T}}$. We then perform singular value decomposition on the matrix formed from $\mathbf{r}_1$, $\mathbf{r}_2$, and the axis orthogonal to the plane spanned by $\mathbf{r}_1$ and $\mathbf{r}_2$, $\mathbf{r}_1 \times \mathbf{r}_2$, to obtain the closest orthonormal rotation matrix to the original affine matrix:

$$\mathbf{U}\mathbf{\Sigma}\mathbf{W}^{\mathrm{T}} = \begin{bmatrix} \mathbf{r}_1 \\ \mathbf{r}_2 \\ \mathbf{r}_1 \times \mathbf{r}_2 \end{bmatrix}. \tag{4.5}$$

The rotation matrix is then given by $\mathbf{R} = \mathbf{U}\mathbf{W}^{\mathrm{T}}$. If $\det(\mathbf{R}) = -1$, the third row of $\mathbf{U}$ needs to be negated and subsequently $\mathbf{R}$ recomputed, to guarantee that it is a valid rotation matrix.

The final result is a full set of scaled orthographic projection parameters $\boldsymbol{\rho}$, consisting of the scale s, rotation matrix $\mathbf{R}$ with orthonormal columns,

and the translation $\mathbf{t}$ containing t_x and t_y. In contrast to other methods which estimate the camera parameters using linear or nonlinear optimisation procedures with constraints, the method used here offers a direct, robust and fast two-step way to estimate a scaled orthographic projection. In practice, together with the iterative fitting approach, we found that it converges fast and provides excellent results, assuming that a face is not severely affected by the perspective effect.

4.2 Closed-form PCA Shape Fitting

Given the estimated camera pose, the 3D shape model is fitted to the sparse set of 2D landmarks to produce an identity-specific 3D shape. To estimate the shape in closed-form solution, we use the linear shape fitting of Aldrian & Smith [AS13]. We find the most likely vector of PCA shape coefficients $\boldsymbol{\alpha}$ by minimising the distance between given 2D landmarks and the projected model points, using the following cost function:

$$\mathbb{E} = \sum_{i=1}^{3L} \frac{(y_{p,i} - y_i)^2}{2\sigma_{2D,i}^2} + \lambda \|\boldsymbol{\alpha}\|^2 , \tag{4.6}$$

where $\|\boldsymbol{\alpha}\|^2$ is a prior on the shape coefficients. L is the number of landmarks, and $\boldsymbol{\sigma}_{2D}^2$ the variances of these landmark points (e.g. obtained during the training of a landmark detector). $\mathbf{y} = [y_1, \dots, y_{3L}]^\mathrm{T}$ is a stacked vector of detected or labelled 2D landmarks in homogeneous coordinates, and $\mathbf{y_p} = [y_{p,1}, \dots, y_{p,3L}]^\mathrm{T}$ is a stacked vector of the 3D Morphable Model shape points that correspond to the respective 2D landmarks, projected to 2D using the estimated camera matrix. To project the model points from 3D to 2D, we construct a camera matrix $\mathbf{C} \in \mathbb{R}^{3\times4}$:

$$\mathbf{C} = s \begin{bmatrix} \mathbf{R}_{1,:} & t_x \\ \mathbf{R}_{2,:} & t_y \\ 0 \;\; 0 \;\; 0 & 1 \end{bmatrix} , \tag{4.7}$$

where $\mathbf{R}_{n,:}$ is the n-th row of $\mathbf{R}$. We then construct a block-diagonal matrix $\mathbf{P} \in \mathbb{R}^{3N \times 4N}$ that contains copies of the camera matrix $\mathbf{C}$ on its diagonal. Finally, we can project all vertices corresponding to the 2D landmarks to the image plane with: $\mathbf{y_p} = \mathbf{P} \cdot (\hat{\mathbf{V}}_h \boldsymbol{\alpha} + \bar{\mathbf{v}})$, where $\hat{\mathbf{V}}_h$ is a modified PCA shape basis matrix that consists only of the rows from the full basis matrix $\mathbf{V}$ that correspond to the landmark points that the shape is fitted to. The basis vectors are multiplied with the square root of their respective eigenvalue, and, because the derivation is expressed in homogeneous coordinates, a row of zeros is inserted after every third row. With this formulation, the cost function in Eq. (4.6) can be expressed in terms of a standard regularised quadratic form with diagonal distance matrix, which has the form:

$$\mathbb{E} = (\mathbf{D}\boldsymbol{\alpha} + \mathbf{b})^{\mathrm{T}} \boldsymbol{\Omega} (\mathbf{D}\boldsymbol{\alpha} + \mathbf{b}) + \lambda \left\| \boldsymbol{\alpha} \right\|^2, \tag{4.8}$$

where we set $\mathbf{D} = \mathbf{P}\hat{\mathbf{V}}_h$, $\mathbf{b} = \mathbf{P}\bar{\mathbf{v}} - \mathbf{y}$, and $\boldsymbol{\Omega} = \mathrm{diag}(\boldsymbol{\sigma}_{2D}^{-2})$.

This standard formulation can be solved in closed-form (derived by [AS13] and in Appendix A), and we arrive at the solution for the shape identity coefficients $\boldsymbol{\alpha}$:

$$\boldsymbol{\alpha} = -(\hat{\mathbf{V}}_h^{\mathrm{T}} \mathbf{P}^{\mathrm{T}} \boldsymbol{\Omega} \mathbf{P} \hat{\mathbf{V}}_h + \lambda \mathbf{I})^{-1} (\hat{\mathbf{V}}_h^{\mathrm{T}} \mathbf{P}^{\mathrm{T}} \boldsymbol{\Omega}^{\mathrm{T}} (\mathbf{P}\bar{\mathbf{v}} - \mathbf{y})), \tag{4.9}$$

where λ is a regularisation parameter that guides the influence of the prior $\|\boldsymbol{\alpha}\|^2$. With the recovered shape coefficients $\boldsymbol{\alpha}$, the corresponding shape instance $\mathbf{S}$ can be generated with Eq. 3.2.

4.3 Linear Expression Fitting

In addition to fitting the PCA identity shape model, we need to estimate facial expressions. We model expressions with a set of additive expression blendshapes $\mathbf{B}$, as introduced in Section 3.3.

To find the blendshape coefficients $\boldsymbol{\psi}$, we use a standard least-squares formulation, similar to Section 4.2. However, since both identity and

expressions have to be estimated, we propose a modified formulation, and combine the identity and expression fitting, which results in a bi-linear system with respect to the unknowns $\boldsymbol{\alpha}$ and $\boldsymbol{\psi}$ that can efficiently be solved by keeping one fixed and estimating the other. The role of the two unknowns is alternated. Instead of using the mean shape $\bar{\mathbf{v}}$ in Eq. 4.9, we substitute it with a face instance $\mathbf{S}^{\boldsymbol{\alpha}}$, generated with the currently estimated $\boldsymbol{\alpha}$:

$$\mathbf{S}^{\boldsymbol{\alpha}} = \bar{\mathbf{v}} + \sum_{i=1}^{m} \alpha_i \sigma_i \mathbf{v}_i. \tag{4.10}$$

We further define the matrix $\hat{\mathbf{B}}_h$, where $\hat{\mathbf{B}}_h$ is modified from $\mathbf{B}$ in the same way as $\hat{\mathbf{V}}_h$ in Section 4.2, and we set:

$$\boldsymbol{\psi} = -(\mathbf{P}\hat{\mathbf{B}}_h)^{-1}(\mathbf{P}\mathbf{S}^{\boldsymbol{\alpha}} - \mathbf{y}). \tag{4.11}$$

We solve this system of equations with a Non-Negative Least Squares (NNLS) algorithm ([LH95]). In practice, we observed that an additional regularisation term (as employed in the shape identity fitting) is not needed in this case. The blendshape coefficients are constrained to be equal or larger than zero by the NNLS algorithm, and they usually do not reach values above around 1 during the fitting. If the fitting is not constrained to negative coefficients, i.e. we were to use the closed-form least-squares solution, we observe implausible shape deformations in a number of images, since the blendshapes are defined as positive vertex offsets. Figure 4.3 shows an example fitting result where we compare the closed-form solution with the solution from the NNLS solver, and we can see implausible mesh deformations, particularly around the mouth region, if the blendshape coefficients are not constrained to be positive.

Once an estimate of the blendshape coefficients $\boldsymbol{\psi}$ is computed, we generate an identity-neutral shape instance $\mathbf{S}^{\psi}$ using these estimated coefficients:

$$\mathbf{S}^{\psi} = \bar{\mathbf{v}} + \sum_{j=1}^{k} \psi_j \mathbf{B}_j, \tag{4.12}$$

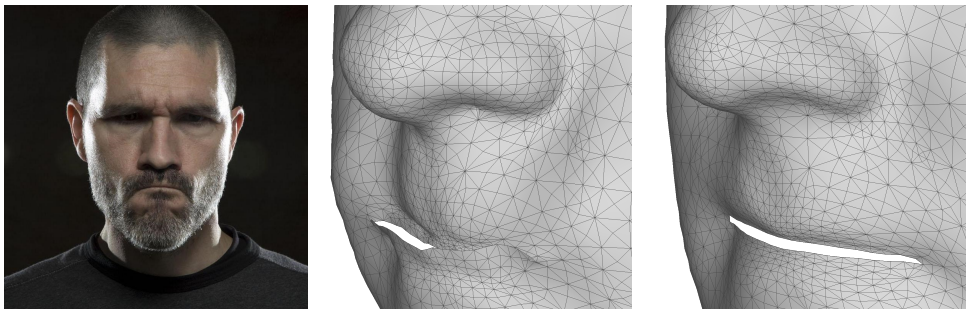


Figure 4.3: Expression fitting with and without the non-negativity constraint. *(left)*: Example image. *(middle)*: Close-up of the resulting mesh with least-squares fitting, without non-negativity constraint. The mesh exhibits implausible deformations. *(right)*: Close-up of the fitted mesh with non-negative least-squares solver. The mesh does not contain implausible deformations.

and use this face instance in Eq. 4.9 of the identity shape fitting instead of the mean face $\bar{\mathbf{v}}$. These two linear systems of the shape identity and expression blendshape fitting can then be solved alternately, with $\mathbf{S}^{\boldsymbol{\alpha}}$ and $\mathbf{S}^{\boldsymbol{\psi}}$ being updated in each iteration. The result of the fitting is the identity-specific shape coefficients $\boldsymbol{\alpha}$ and expression blendshape coefficients $\boldsymbol{\psi}$.

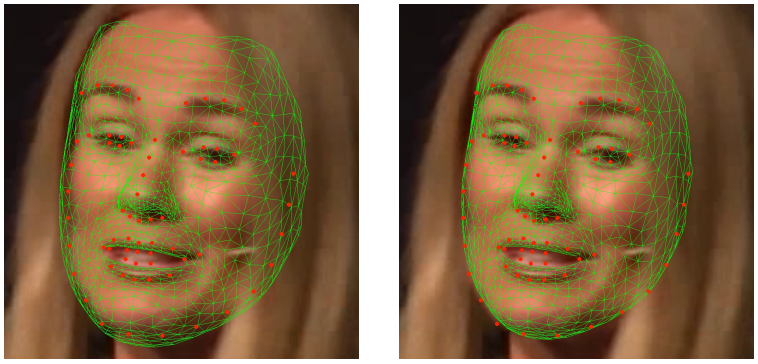
Besides modelling the subject's expressions, blendshape fitting can be used to remove a facial expression from a subject, or to re-render it with a different expression. Figure 4.4 shows an example video frame with a strong expression, the expression-neutralised face, and a re-rendering with a synthesised expression.

Figure 4.4: Frame with strong expression and expression-neutralised image. *(left)*: Input frame. *(middle)*: Expression-neutralised 3D model. *(right)*: Face with artificially added smile expression.

4.4 Dynamic Contour Correspondences

In general, the outer face contours present in the 2D image do not correspond to unique contours on the 3D model. At the same time, these contours are important for an accurate face reconstruction, as they define the boundary region of the face. This problem has had limited attention in the research community, but for example Bas et al. [BSBW16] recently provided an excellent overview describing the problem in more detail.

Figure 4.5 demonstrates the problem and importance of contour fitting, and the result with the proposed algorithm. Apart from being important for an accurate shape fitting, this also has a direct impact on the texture extraction. If the model is not fitted well around the face contour, the extracted texture will contain a significant amount of background.



(a) Without contour fitting (b) With the proposed contour fitting

Figure 4.5: Importance of the facial contour for accurate 3D shape and appearance recovery.

To deal with this problem of contour correspondences, we introduce a contour fitting approach consisting of two separate components. Given a current pose estimate (for example obtained using all non-contour landmarks), the 2D contour is separated into the front-facing (camera-facing) contour, and the back-facing, occluding edge contour. Figure 4.6 *(left)* depicts the front-facing and occluded contours. They are then fitted separately, as

described in the next sections.

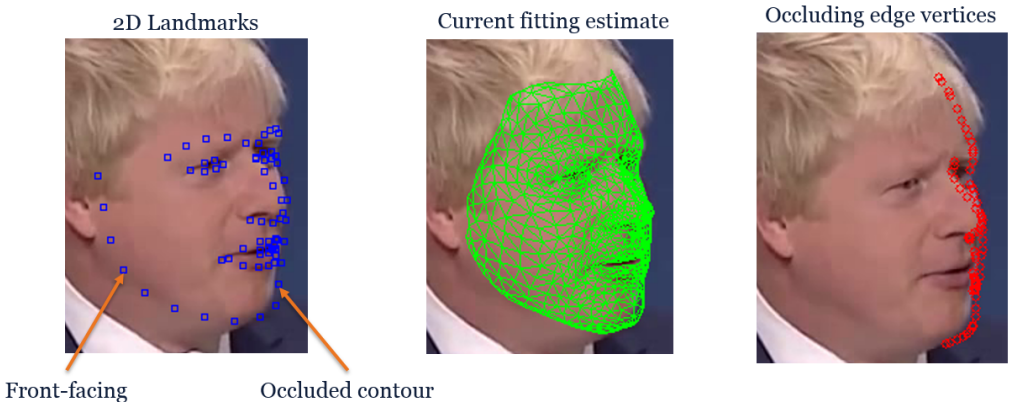


Figure 4.6: Overview of the proposed occluding-contour fitting. *(left)*: All detected 2D landmarks. *(middle)*: 3D mesh of the current 3D fitting estimate, projected to 2D. *(right)*: The mesh vertices satisfying our occluding boundary requirement.

4.4.1 Front-facing Contour

Given the topology of the Surrey Face Model, the front-facing contour (that is, the half of the contour closer to the camera, for example the right face contour when a subject looks to the left) approximately follows the outline of the mesh from the chin, alongside the neck and up to the ears. We can thus fit the front-facing face contour by using semi-fixed 2D–3D correspondences from a list of candidate points along the mesh outline. We define the set of vertices $\mathcal{V}$ along the outline of the 3D face model. Given an initial fit, we then search for the closest vertex in that list for each detected 2D contour point. For a particular 2D contour landmark y, the optimal corresponding 3D vertex $\hat{\mathbf{v}}$ is chosen as:

$$\hat{\mathbf{v}} = \arg\min_{\mathbf{v}\in\mathcal{V}} \|\mathbf{C}\mathbf{v} - y\|^2 , \tag{4.13}$$

where $\mathbf{C}$ is the currently estimated projection matrix from 3D to 2D. As our used morphable model consists of a relatively low-resolution mesh, $\mathcal{V}$ is small, and we can find $\hat{\mathbf{v}}$ by computing all distances and then selecting the one with the minimum distance.

Using a whole set of potential 3D contour vertices makes the method robust against varying roll and pitch angles. It also makes the method robust against vertical inaccuracies of the contour from the landmark regressor, since the contour landmarks of 2D landmark regressors are usually not clearly defined. Once found, these contour correspondences are then used as additional corresponding points in the subsequent fitting steps.

4.4.2 Occluding Contour

The occluded contour is more difficult to fit, as it is not possible to pre-define the corresponding set of 3D vertices for the 2D landmarks; the 2D contour marches along the mesh, as the pose changes, and the corresponding points change with them. Therefore, we introduce an algorithm that dynamically selects the corresponding 3D vertices, given the current pose and shape parameter estimates.

First, as we want to map the occluding 2D contour to the occluding 3D contour, we generate a set of all possible occluding edge vertices (candidates), given the current fitting estimate. These vertices, and their positions in 2D, are depicted in red in Figure 4.6 *(right)*. To compute these, we find all the mesh edges where the normals of the adjacent two triangles are positive in one and negative in the other triangle — which is the definition of an occluding edge. To facilitate a fast lookup of the triangles that are adjacent to each edge, we precompute the edge topology for the mesh once and store it. Then, for each so-found occluding mesh edge, we add the two vertices comprising the edge to the list of occluding edge vertices, and remove duplicates.

Additionally, we need to compute whether each vertex is visible, or occluded by another triangle. We do so by ray-casting from the camera origin to each vertex, and checking if any triangle (other than the triangles that the target vertex is part of) intersects the ray. Self-occluded vertices

are removed as well.

During the fitting, for each 2D contour landmark, we then find the closest (in a l^2 sense) occluding edge vertex from the list of all computed occluding edge vertices. This is done by building a k-d tree of the occluding edge vertices, which then makes a look-up for a given 2D landmark point an efficient binary search (see e.g. Muja & Lowe [ML14]). The so-found new correspondences are added to the list of 2D–3D correspondences, used in subsequent fitting steps (and these correspondences are re-computed in each iteration of the fitting).

4.5 Recapitulation

The flowchart in Figure 4.7 summarises the full proposed real-time shape-to-landmarks fitting algorithm. The fitting is initialised by computing a rough pose estimate using only the inner face landmarks (i.e. excluding all contour landmarks), followed by an initial estimate of the expressions. Then, both components of the contour fitting process are applied to get additional correspondences for both the front-facing as well as the occluded face contour. Subsequently, the pose is re-estimated using all landmarks, including the contour landmarks, followed by solving for PCA shape identity coefficients and blendshapes. These four main components are iterated towards convergence.

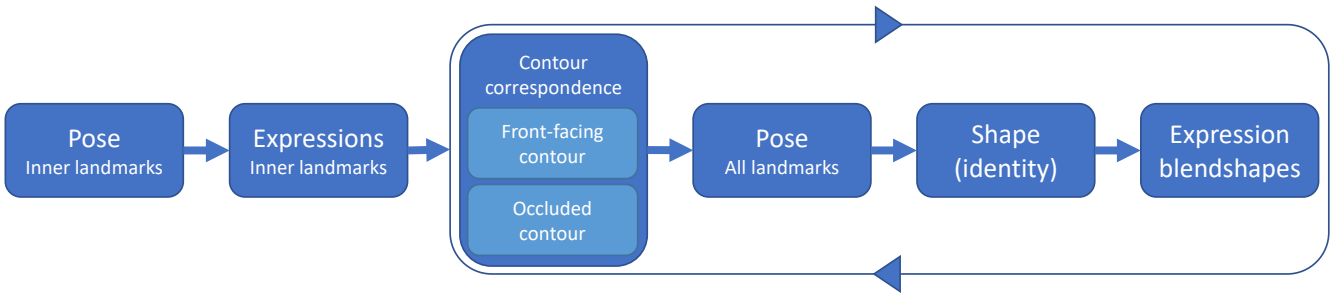
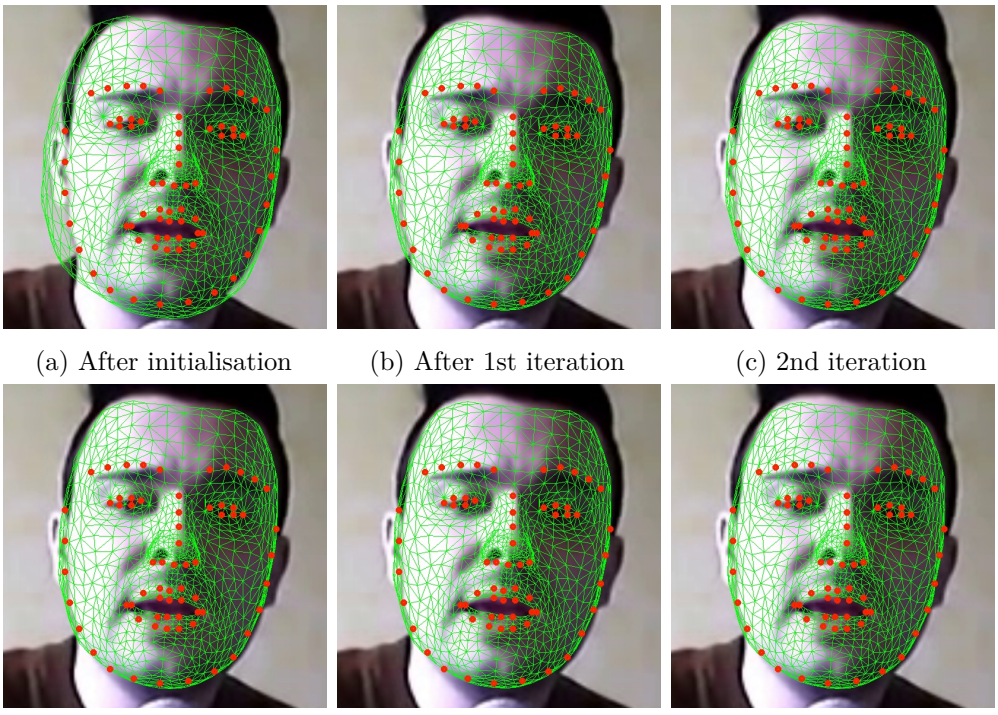


Figure 4.7: Flowchart of the iterative linear fitting algorithm. The fitting is initialised with a rough pose and expression fit, and then proceeds alternating contour, pose, identity and expression fitting.

4.6 Convergence

First, we analyse the convergence of the fitting on example videos. Figure 4.8 shows a typical example of a fitting result after various number of iterations. The first picture shows the fitting estimate after only the initial pose and expression fit with inner landmarks. The second picture shows the result after one complete iteration, i.e. each fitting step was performed once. The subsequent pictures show the fitting results after the 2nd, 5th, 10th, and 50th iteration. It can be observed that the fitting quality increases notably from the initialisation to performing a full iteration. The subsequent few iterations are beneficial, and the fitting converges fast up to the 5th iteration. After that, hardly any difference can be observed in the fitted mesh.



(a) After initialisation (b) After 1st iteration (c) 2nd iteration

(d) 5th iteration (e) 10th iteration (f) 50th iteration

Figure 4.8: Fitting result after various number of iterations. The fitting improves significantly after the first and second iteration. After around 5 to 10 iterations, there are no longer visible differences in the fitted mesh.

We further study the convergence of the camera parameters and the

shape and expression coefficients over the course of iterations. Figure 4.9 shows the values of the camera parameters, and Figure 4.10 the values of the first 10 shape and the 6 expression blendshape coefficients, over 75 fitting iterations, on an exemplar image. It can be seen that all the parameters fully converge, though requiring more iterations than is visible in the previous figure, Figure 4.8. We observe the same convergence behaviour on all images that we tested on.

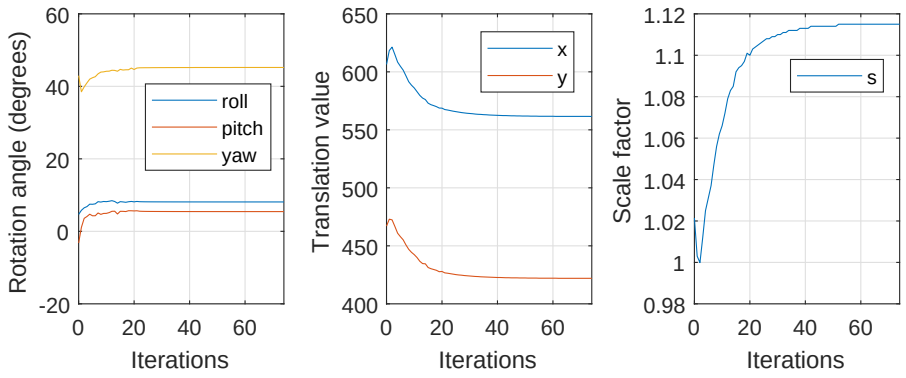
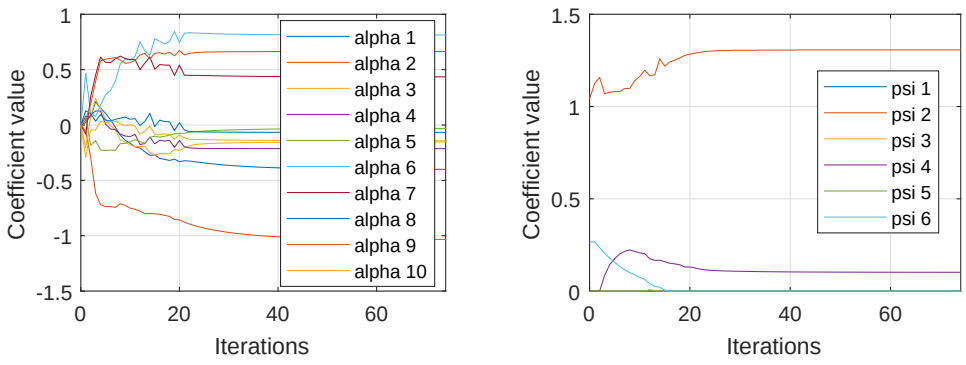


Figure 4.9: Convergence of the camera parameters (rotation, translation and scale) over the course of 75 iterations.



(a) (b)

Figure 4.10: Convergence of *(a)* the first 10 PCA shape parameters and *(b)* the 6 used expression blendshape coefficients, over the course of 75 iterations.

To verify how much the actual reconstructed mesh changes in each fitting iteration, we investigate how the x, y and z positions of three hand-chosen

vertices change. We chose the tip of the nose (vertex 114), right corner of the mouth (vertex 398), and a random point on the left cheek (vertex 707). Figure 4.11 shows the x, y and z coordinates of these points over 75 fitting iterations. We can see that after 10 to 20 iterations, all three vertices stay stable and their 3D position does not change anymore. This finding confirms our observation from Figure 4.8 that after 5 to 10 iterations, the 3D mesh no longer noticeably changes.

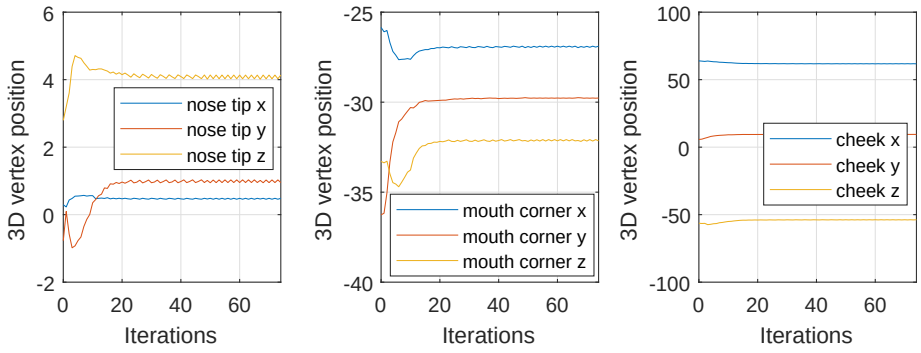


Figure 4.11: Convergence of three different 3D mesh positions, over the course of 75 iterations: the tip of the nose, right mouth corner, and a point on the left cheek.

4.7 Run Time

In this section, we analyse the total run time of the proposed real-time shape fitting algorithm as well as the run time of the individual components. We measure the run time of the pose, shape, expression and contour fitting separately, using both the 845 vertices 3DMM as well as the 3448 vertices model. The measurements were done on a Core i7-7700HQ CPU, with no efforts made to parallelise the code. The numbers were measured on a video of 150 frames length, where a subject changes their yaw and pitch pose angle as well as their facial expressions significantly over the course of the video. Table 4.1 provides an overview of the resulting run time, measured with 68 facial landmarks, and fitting all principal components of

the shape model. In all the cases, the run time of each of the components is of the order of a few hundred microseconds, and the total run time for one complete iteration with the 845 vertices model is below 1 millisecond. The proposed fitting algorithm is thus able to achieve above 1000 frames per second (fps) in this scenario.

The standard deviation shows that the time consumed by the fitting varies between frames. This is caused mainly by the dynamic contour fitting, as the number of landmarks used differs from frame to frame. Some of the fitting components are affected more by this, for example the contour fitting. In general, with the 3448 vertices model, the contour fitting is comparatively time-consuming: the computation of the occluding vertices involves ray-casting to determine whether a vertex is self-occluded. If necessary, this could further be speeded up by employing more advanced ray-casting techniques.

Table 4.1: Run time of each component of the fitting algorithm (mean and standard deviation over an exemplar video), in microseconds.

	3448 vertices model	845 vertices model
Pose estimation	232 µs ± 82	129 µs ± 32
Shape fitting	466 µs ± 151	404 µs ± 91
Expression fitting	335 µs ± 125	82 µs ± 26
Contour correspondences	1486 µs ± 516	362 µs ± 112
Total	**2519 µs**	**977 µs**

With a run time this low, several fitting iterations can be easily performed, while still achieving real-time performance. For example, running the algorithm for 5 iterations, which the previous section determined to be an often-sufficient number, results in a run time of 200 fps with the 845 vertices model and 80 fps with the 3448 vertices model.

The run time can further be reduced by refraining from fitting all available

principal components. By only estimating the first 10 shape coefficients, the run time of the shape fitting part reduces to approximately one third of the reported numbers. Estimating more coefficients when fitting to landmarks on a single image is not necessarily beneficial, as the further components represent mainly noise. Additionally, we observed that the run time of the pose, shape and expression fitting depends approximately linearly on the number of landmarks used for the fitting.

These high frame rates, achieved on a regular notebook CPU, without using any GPU acceleration, give rise to a number of applications, including running on mobile devices, where run time as well as power consumption are large concerns.

4.8 3D Reconstruction Accuracy

The proposed fitting algorithm is subsequently evaluated in various ways and compared against a state-of-the-art algorithm. We first evaluate the 3D shape reconstruction accuracy of the single-image fitting. It is a challenging problem in itself to evaluate such algorithms in an in-the-wild scenario. Datasets which contain in-the-wild images, alongside 3D ground truth, are hard to come by as it is a requirement quite hard to meet. Data with accurate 3D ground truth is usually captured in labs with 3D scanning devices. On the other hand, regular cameras capture in-the-wild images, but there is no 3D ground truth available. One of the few available datasets meeting these requirements is the AFLW2000-3D dataset from Zhu et al. [ZLL+16], used in our experimental evaluation.

We compare the 3DDFA algorithm, published by the same authors ([ZLL+16]), to our approach on the AFLW2000-3D dataset. The dataset contains the first 2000 images of the original AFLW dataset (Köstinger et al. [KWRB11]), alongside 2D landmarks in the ibug 68 facial points mark-

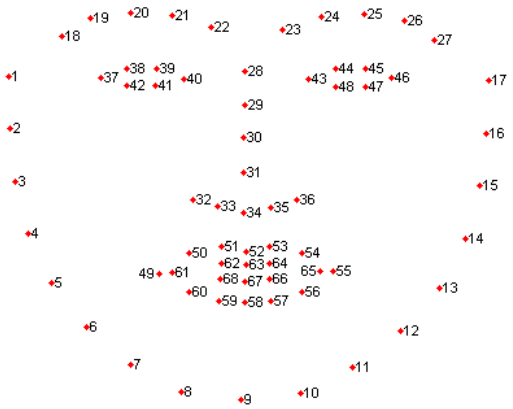


Figure 4.12: The ibug 68 facial landmark points mark-up.

up (see Figure 4.12 and Sagonas et al. [SAT+16]). The images are gathered from Flickr, and exhibit a large variation in pose (up to profile), expressions, and face shape. AFLW2000-3D also contains what the authors refer to as *ground truth* 3D shapes. In practice, the 3D *ground truth* is created with a variant of the Multi-features fitting (MFF, Romdhani and Vetter [RV05]) algorithm, initialised with the 2D landmarks from the dataset, and ran with a modified version of the Basel Face Model that additionally contains a PCA expression basis. As the original MFF fitting has been shown not to be very accurate on in-the-wild images (see e.g. [BAP+17a]), we advise to be cautious in taking these MFF fitting results as the ground truth for evaluation. However, the authors of 3DDFA seemed to have used an improved version of MFF which results in decent quality output overall, and we proceed with the comparison on that basis. Also, the authors note that the results are verified by them and manually corrected where necessary. In contrast to the original paper, we evaluate the actual 3D shape reconstruction accuracy, not the landmark error reprojected to 2D. In our opinion, this provides a more principled evaluation metric, as the accuracy of the depth reconstruction is an important component of 3D face reconstruction.

We fit the Surrey Face Model with 3448 vertices resolution to each image in the dataset, using the 68 landmarks provided by the authors. We set the regularisation parameter λ of the PCA shape fitting to 30 and run the fitting for 10 iterations. To evaluate the 3D vertex error for our approach, we register the fitted mesh from the Surrey Face Model to the ground truth mesh (which is a modified BFM mesh topology) using the non-rigid ICP approach of Medina et al. [AAB+14] in the Menpo framework, which is a reimplementation of Amberg et al. [ARV07]. This results in a representation of the ground truth mesh with the topology of the Surrey Face Model. We then compare the mesh from the fitting result with the registered ground truth mesh by using the distance from each vertex of the ground truth to the nearest point on the mesh of the fitting result, over the face region. The vertex error is normalised with the outer-eye-distance of each mesh.

To produce the results for 3DDFA, we run it with default settings and the same 68 landmarks, and compute the vertex error directly from the ground truth 3D meshes to their fitting results (as they already have the same topology), over the same face region.

We then plot a curve for each algorithm showing the cumulative distribution of the error from all vertices and all images with respect to the total number of measurement points. Figure 4.13 shows the results of that comparison. It can be seen that the proposed fitting approach slightly outperforms 3DDFA in 3D face reconstruction accuracy. It is remarkable to note that our shape-to-landmarks fitting, which does not refine shape using image information, outperforms a much more complex algorithm. The same trend will be observed in the later evaluation of the multi-image fitting (Section 5.2). Additionally, the 3DDFA method is learning-based, and thus it is unclear how well it generalises to a new dataset, while our approach solely depends on landmarks, which can be obtained robustly by

any state-of-the-art landmark detection algorithm.

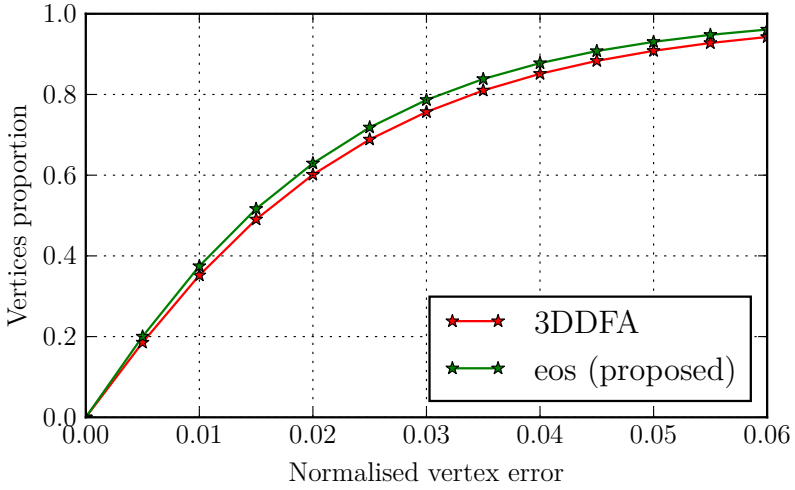


Figure 4.13: 3D shape reconstruction error of the proposed single-image fitting compared to 3DDFA [ZLL+16] on AFLW2000-3D.

In terms of run time, the proposed algorithm is faster throughout: 3DDFA report a total run time of around 100 ms with a powerful CPU *and* GPU if all steps of their algorithm are summed up, while our proposed algorithm took around 25 ms on a CPU.

Figure 4.14 shows a number of example fitting results on the AFLW2000-3D dataset. We can observe that even fitting to landmarks from full profile images works very well.

4.9 Qualitative Evaluation

To further demonstrate the capabilities of the proposed shape-to-landmarks fitting, it was run on a second dataset, the HELEN test set [SAT+16, LBL+12], another dataset with a number of images with extreme pose and expressions. The fitting was run with the 68 ibug landmarks, and the same parameters than for the images on AFLW2000-3D. In this case, no 3D ground truth exists, and we thus only show qualitative results on the dataset.

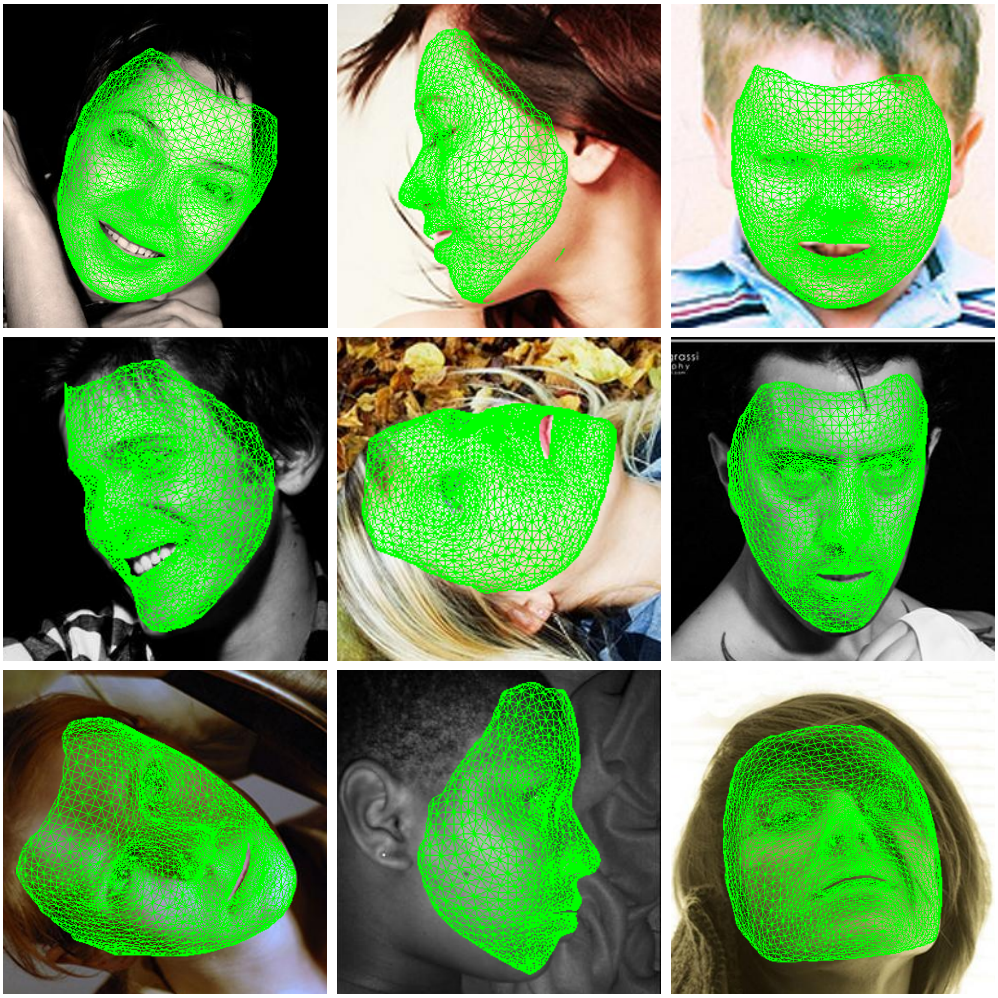


Figure 4.14: Example images of AFLW2000-3D with rendering of the fitted 3D mesh overlaid as wireframe.

Figure 4.15 shows example results from the HELEN testset. The fitting results shown give an overall impression of the fitting quality on the whole dataset, and are not hand-picked best results. In fact, we found that the algorithm did not completely fail on any of the images. In the bottom row, we show a few hand-picked examples of the worst fitting results on the dataset. These show only small artefacts, most visible around the chin, the ear-region, and on the forehead, where the shape starts to deform in implausible ways. Even in these cases the algorithm is able to recover an overall convincing shape. In general, we can see that the algorithm is able to cope well with a variety of faces like children's faces, especially wide or narrow faces, and even strong and asymmetrical facial expressions.

4.10 Summary

This chapter presented three key contributions. First, we presented a linear solution for facial expression blendshapes fitting, based on the closed-form PCA shape identity fitting of Aldrian & Smith [AS13], which, both together, result in a system that is bi-linear in the expression and identity coefficients, and is efficiently solved by alternating between the sets of identity and expression parameters being estimated. Second, we proposed a robust and fast contour fitting approach to dynamically determine the 2D–3D correspondences between detected contour landmark points and 3D model points. Third, the combination of all steps result in a robust shape-to-landmarks fitting algorithm that is able to deal with the large number of challenges in in-the-wild images, including large poses and strong asymmetrical expressions. We thoroughly evaluated the proposed algorithm and showed that the landmark-fitting approach delivers state of the art performance. By having designed most components with a linear cost function, the algorithm achieves a run time of up to 1000 fps, opening the door to a variety of novel applications.

Figure 4.15: Example images of the HELEN testset with rendering of the fitted 3D mesh overlaid as wireframe. The algorithm adapts well to faces of various origin like children's faces or wider faces, and it even copes well with strong and asymmetrical expressions. *(Bottom row)*: Hand-picked worst examples.

Chapter 5

Multi-frame Fitting

When reconstructing a 3D face from multiple images or frames of a video, the trivial strategy is to fit the model on a frame-by-frame basis, and then blend or merge the individual meshes. However, this may not be the most ideal strategy, even if we separate face shape from expressions, which can be different in each image. If it is known that all images or frames contain observations of the same identity, it would be useful to incorporate this knowledge as part of the fitting algorithm. Furthermore, to build a fully textured 3D face model of a particular person, the appearance information from the multiple images needs to be combined.

In the following, we will extend the proposed shape fitting algorithm to the scenario of fitting to multiple images of the same identity as well as video sequences, making use of the temporal coherence of the identity within a set of frames. We further devise an efficient strategy to model face appearance from in-the-wild data by fusing image texture information from these multiple images.

5.1 Multi-frame Shape Fitting

First, we extend the real-time shape-to-landmarks fitting approach introduced in Chapter 4 to multiple images. Given a video or multiple images of the same person, they all are observations of the same shape. Thus

the fitting of the 3D face shape model to multiple frames involves the recovery of a single set of shape parameters, given all images. Similar to Section 4.2, when the pose and expression of each frame are given, and the scaled orthographic projection is used, we can recover the identity PCA shape coefficients jointly for all frames with a closed-form solution.

As in Section 4.2, we set up a linear system of equations — only in this case, each image contributes to the total number of equations, depending on how many landmarks are used in the image. For each of these landmarks, the camera matrix and expression shape estimates corresponding to the specific image are used. We assemble a block-diagonal matrix $\mathbf{P}^{*}$[2] consisting of the camera matrices $\mathbf{P}_i$ for each frame i (see Section 4.2) on its diagonal:

[2] The asterisk denotes the multi-image versions of each of the previously used matrices.

$$\mathbf{P}^{*} = \begin{bmatrix} \mathbf{P}_1 & & \\ & \ddots & \\ & & \mathbf{P}_n \end{bmatrix}, \tag{5.1}$$

where n is the total number of images or frames.

The modified PCA basis matrix $\hat{\mathbf{V}}_h$ from the single-image fitting in Section 4.2 is constructed by vertically stacking the matrices from the individual images:

$$\hat{\mathbf{V}}_h^{*} = \begin{bmatrix} \hat{\mathbf{V}}_{h,1} \\ \vdots \\ \hat{\mathbf{V}}_{h,n} \end{bmatrix}. \tag{5.2}$$

The mean-matrix $\bar{\mathbf{v}}^{*}$ for the multi-image fitting is constructed in a similar way, by stacking the mean face $\bar{\mathbf{v}}$ n times. The 2D landmarks vector $\mathbf{y}^{*}$ is assembled by stacking all 2D landmark positions $\mathbf{y}_i$ of all n images one after another.

We then recover the identity coefficients $\boldsymbol{\alpha}$, taking into account all landmarks from all images, using the same regularised quadratic form from

Section 4.2:

$$\boldsymbol{\alpha} = -(\hat{\mathbf{V}}_h^{*\mathrm{T}}\mathbf{P}^{*\mathrm{T}}\boldsymbol{\Omega}^*\mathbf{P}^*\hat{\mathbf{V}}_h^* + \lambda\mathbf{I})^{-1}(\hat{\mathbf{V}}_h^{*\mathrm{T}}\mathbf{P}^{*\mathrm{T}}\boldsymbol{\Omega}^{*\mathrm{T}}(\mathbf{P}^*\bar{\mathbf{v}}^* - \mathbf{y}^*)), \tag{5.3}$$

where $\boldsymbol{\Omega}^* = \mathrm{diag}(\boldsymbol{\sigma}_{2D}^{-2})$, with one value for each landmark on each frame.

To initialise the fitting, the camera pose $\mathbf{P}_i$ for each image i is computed, followed by an initial estimate of the expressions in each image (as expressions and poses can vary between multiple images). Then the contour fitting introduced in Section 4.4 is applied to each image, followed by re-estimating the pose using all landmarks, including the contour landmarks. Finally, we solve for the identity coefficients, given the pose and expression estimates of all images. The overall process is similar to the single-image fitting of Chapter 4, which was summarised in Figure 4.7. The multi-frame fitting is iterated for a number of iterations until convergence to a stable solution. It is detailed in Algorithm 1.

5.2 Evaluation of Shape Reconstruction Accuracy

To evaluate the accuracy of the proposed multi-image fitting and compare it to other approaches run on in-the-wild imagery, we acquired a subset of the KF-ITW database from Booth et al. [BAP+17a], kindly made available to us by the authors. The database contains the 2D videostream of recordings made with a Kinect v1 camera, captured under relatively unconstrained conditions, where the camera is moved around the subject from left to right, from a pose of around -30° yaw angle to +30°. Each video is supplied with automatically detected 2D landmarks in the ibug 68 points mark-up, and a 3D ground truth mesh of the subject, recovered from the whole video with KinectFusion ([NIH+11]). The subset of the database available to us, which we will refer to as KF-ITW-pr (prerelease), contains data of 5 people (out of 17 in total), each in neutral expression as well as either a happy or surprise expression (or both for some subjects), together with the 3D

input : n sets of landmarks $\{\boldsymbol{y}\}_n$ from n images

output : n sets of pose parameters $\{\boldsymbol{\rho}\}_n$ and expression coefficients $\{\boldsymbol{\psi}\}_n$; one set of identity coefficients $\boldsymbol{\alpha}$

for $i = 1$ **to** n **do** `// Initial pose and expression fit`

  estimate the scaled orthographic projection $\{\boldsymbol{\rho}\}_i$;

  estimate the expression blendshape coefficients $\{\boldsymbol{\psi}\}_i$;

end

for $k = 1$ **to** *num_iterations* **do**

  for $i = 1$ **to** n **do**

    retrieve the 2D–3D correspondences for the inner face landmarks;

    given the current $\{\boldsymbol{\psi}\}_i$ and $\{\boldsymbol{\rho}\}_i$:

    - find correspondences of the front-facing face contour;
    - find the occluding contour vertices of the detected contour landmarks;

    re-estimate the pose parameters $\{\boldsymbol{\rho}\}_i$, using all correspondences;

  end

  `// Fit the identity from all frames:`

  estimate $\boldsymbol{\alpha}$ from all $\{\boldsymbol{\psi}\}$ and $\{\boldsymbol{\rho}\}$;

  `// Estimate expressions in each frame:`

  for $i = 1$ **to** n **do**

    estimate the expression coefficients $\{\boldsymbol{\psi}\}_i$;

  end

end

Algorithm 1: Multi-frame shape fitting.

ground truth. The videos are between 60 to 180 frames in length. Figure 5.1 shows a frame of one of the KF-ITW videos, and the corresponding 3D ground truth.

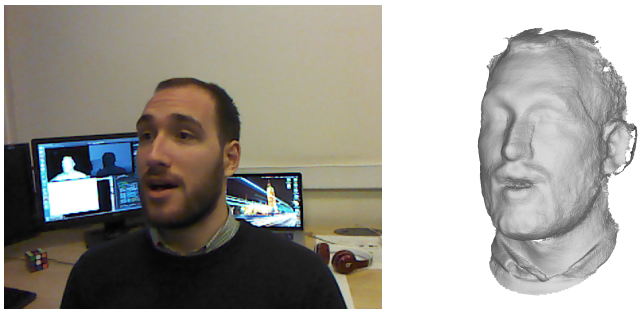
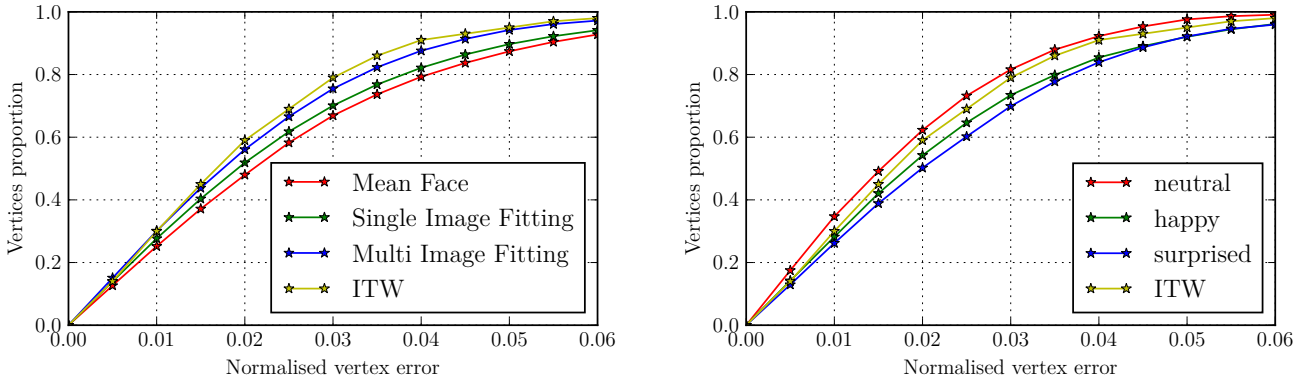


Figure 5.1: Example frame and 3D ground truth from the KF-ITW dataset.

We follow the evaluation protocol of Booth et al. [BAP+17a] on the database. First, we register the Surrey Face Model mesh to the ground truth 3D scan of each video with the non-rigid ICP approach of Medina et al. [AAB+14] in the Menpo framework. This is the same algorithm as Booth et al. used in their evaluation. We then fit the 3448 vertices Surrey Face Model to each video, using the proposed multi-frame fitting approach, with the same parameters as used in Section 4.8. The model is fitted to the 68 landmarks that come with the database. For each video, we subsequently compare the fitting result to above registered ground truth scan by measuring the distance from each vertex on the scan to the closest point on the mesh of the fitting result. The evaluation is done on a mask that covers the face area and is slightly smaller than the mesh of the Surrey Face Model — it is the same mask as used by the authors of KF-ITW. The vertex error is normalised with the outer-eye-distance of each scan. We then plot curves showing the cumulative distribution of the error from all vertices and all frames with respect to the total number of error measurements.

Figure 5.2a shows the results of our multi-image fitting on the KF-ITW-pr subset compared to the results from Booth et al. [BAP+17a] on the full

database. Additionally, we measure the accuracy when fitting using only a



(a) (b)

Figure 5.2: Results of our multi-image fitting on a subset of the KF-ITW database. *(a)*: Comparing our multi-image fitting with single-image fitting, fitting using only a mean face, and the ITW fitting method of [BAP+17a]. *(b)*: Comparing our performance on different expression subsets. Overall, our multi-image fitting performs comparably to [BAP+17a], while being significantly simpler.

3D mean face, and the accuracy of using per-frame single-image fitting. We observe that performing single-image fitting is better than using just the mean face, and multi-image fitting provides a significant benefit to single-image fitting. The ITW fitting approach [BAP+17a] slightly outperforms our method. It is noteworthy that they do not give a run time for their algorithm. We can however estimate that their algorithm would most likely require of the order of seconds or minutes, as it solves a complex nonlinear optimisation task using the Gauss-Newton method.

In addition to that, it was confirmed to us by the authors that our subset of the KF-ITW database contains proportionally more videos with expressions than the whole database, thus potentially putting algorithms evaluated on only the subset at a disadvantage, if fitting to neutral faces is easier than fitting to faces with expressions. To investigate this, we evaluated the neutral, happy and surprised expressions separately for our multi-image fitting. Figure 5.2b shows the results on the separate subsets. It can be seen that fitting on the expression-neutral videos results in the

highest performance, and on this particular subset, the proposed algorithm outperforms the ITW fitting approach ran on the whole database. We believe that videos with expressions are harder to fit because the expressions add deformations on top of a neutral face, which must be accounted for and can introduce additional errors. We thus conclude that on the full database, containing fewer videos with expressions, it is likely the performance of our algorithm would be slightly higher than shown in Figure 5.2a. Overall, our algorithm performs comparatively to the ITW fitting algorithm while being considerably simpler by not relying on a learning-based approach and nonlinear optimisation, and we reckon it to be much faster.

In an independent evaluation, performed by Booth et al., the performance of an earlier version of the proposed algorithm, with single-image fitting, was measured. Figure 5.3 shows the results. The earlier version of the proposed algorithm performs nearly identical to the proposed single image fitting in Figure 5.2a, validating our experimental protocol. Additionally, it can be seen that the proposed algorithm clearly outperforms "classic" 3DMM fitting based on the nonlinear Multi-features-fitting (MFF) initially proposed by Romdhani and Vetter [RV05]. The curves with asterisk (*) are taken from the paper of Booth et al. [BAP+17a].

5.3 Analysis and Convergence

While the camera and expression parameters we estimate in the fitting process are image specific, the shape parameters are identity specific and constant within one video (given the same person is followed throughout the whole video). In the following, we analyse the convergence of the shape parameters depending on the number of fitting iterations and the number of images used for fitting.

First, we investigate the number of images needed to obtain a stable

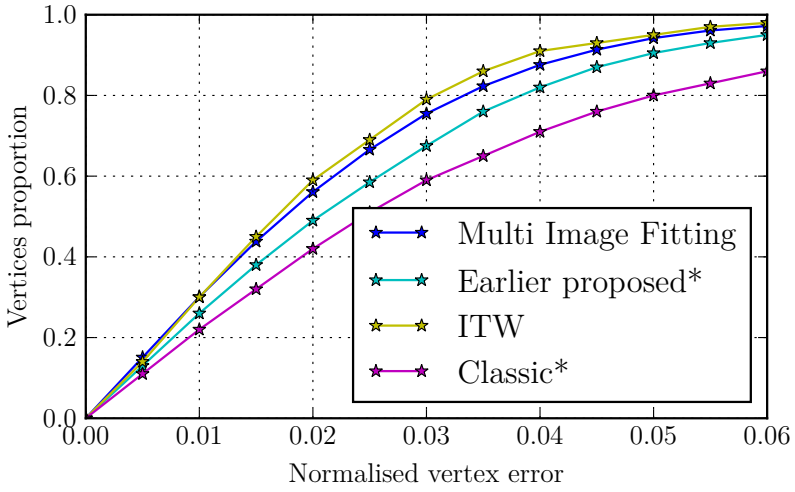


Figure 5.3: Comparison of the proposed method to "classic" 3DMM fitting, and independent evaluation of an earlier version of the proposed algorithm. The curves with asterisk (*) are taken from the paper of Booth et al. [BAP+17a].

identity representation for a given subject. In Figure 5.4, we plot the deviations of the shape coefficients, given different numbers of images. For each number (i.e. point n on the x-axis), we draw random sets of n images from the total set of frames, and run the proposed multi-frame shape fitting on each set. We then compute the standard deviations of the PCA identity coefficients $\boldsymbol{\alpha}$ over the sets. As the identity is constant throughout a video, a low standard deviation means that fitting to the different random sets of images drawn from all frames resulted in similar shape coefficients. The plot shows the mean of these deviations and their standard deviation over all videos in KF-ITW-pr. We can see that with an increasing number of images used for the fitting, both the mean deviations decrease (i.e. the coefficient values become more stable) as well as the standard deviation itself decreases (i.e. the variations of the coefficients within one video with various subsets decrease). With around 10 to 20 images, the shape coefficients can be estimated quite robustly, independently of the subset of frames that is chosen from a particular video.

With respect to run time, we observe in our experiments that the run

time increases linearly with the number of images used for the multi-image fitting. In preliminary experiments, we were thus able to run the proposed algorithm in real-time on a live video stream (see Section 7.1).

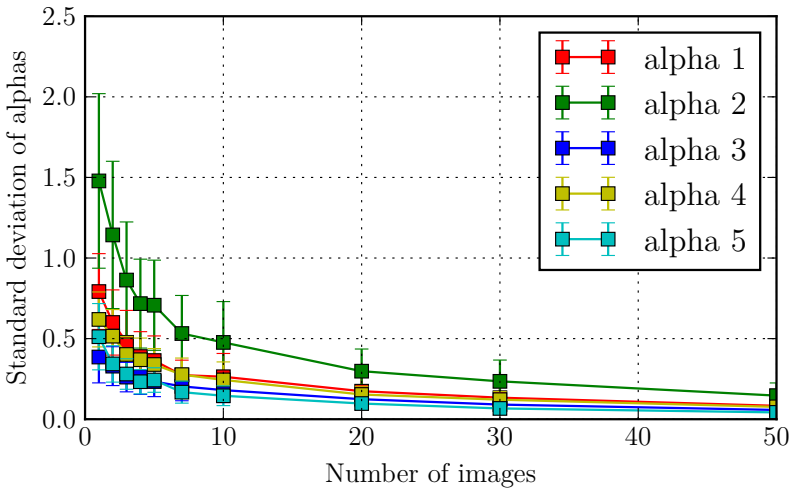


Figure 5.4: Mean deviation of the first five PCA identity coefficients, using various number of images for multi-frame fitting. The error bars correspond to the standard deviation over all measurements for a given data point.

Second, we investigate the convergence of the model parameters over the course of the iterations. Figure 5.5 shows the first 10 shape coefficients for an exemplar video of KF-ITW. The fitting starts with the mean face as initial estimation, where all shape coefficients are 0. The shape coefficients then become more distinct and converge within 50 to 70 iterations. This is consistent with the observations from Section 4.6.

Last, we evaluate the shape reconstruction accuracy with respect to the number of iterations that the multi-image fitting algorithm is run. Figure 5.6 shows the 3D mesh reconstruction accuracy with the mean face, and after 1, 3, 5 and 10 fitting iterations. The results further confirm our observations from Section 4.6, and specifically Figures 4.8 and 4.11: while the actual parameters require more iterations to fully converge, a stable 3D mesh is recovered after around 5 iterations, with the recovered mesh not changing significantly any more thereafter.

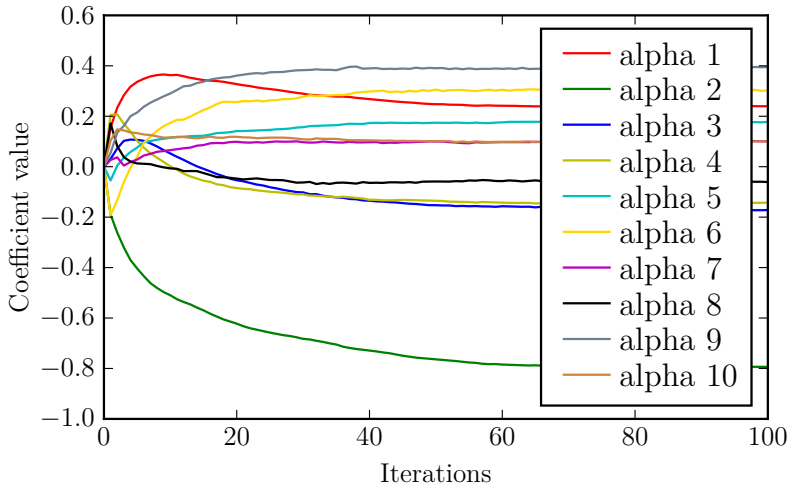


Figure 5.5: Convergence of the shape coefficients over the number of fitting iterations, on an exemplar video.

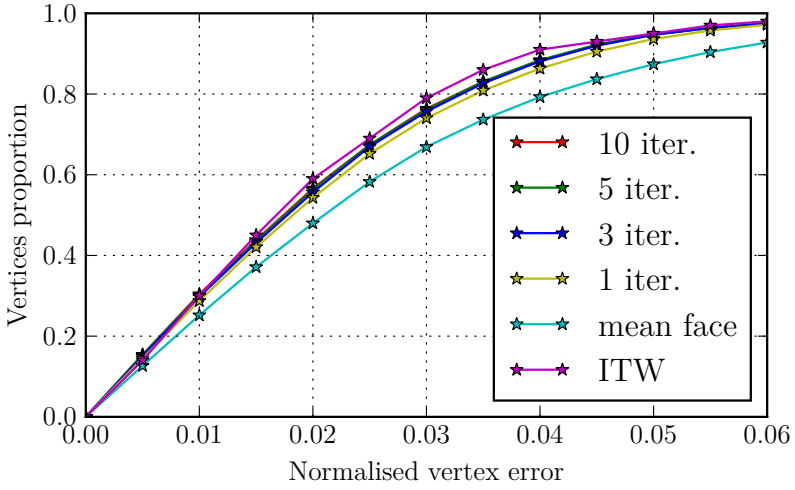


Figure 5.6: Mesh reconstruction accuracy on KF-ITW with different numbers of iterations. The accuracy increases when running for more iterations, and saturates at around 5 to 10 iterations. The reconstruction accuracy with just the mean face is given for comparison.

5.4 Texture Reconstruction

Once an accurate shape model fit is obtained, we reconstruct the appearance of a given face. Since we abstain from using the PCA colour model, we remap the image texture from a frame to an isomap that puts each pixel into a globally registered representation. The isomap is a texture map, created by projecting the 3D model's triangles to 2D while preserving the geodesic distances between vertices (Tenenbaum et al. [TSL00], Rodríguez [Rod07]). The mapping is computed only once, so the isomaps of all of the frames are in dense correspondence with each other. Note that the texture map resolution is independent of the number of vertices N of the shape model.

Inspired by Van Rootseler et al. [vRSV12], we compute a weighting ω for each point in the isomap that is given by the cosine of the angle of the camera viewing direction $\mathbf{d}$ and the normal $\mathbf{n}$ of the 3D mesh's triangle that corresponds to the point: $\omega = \langle \mathbf{d}, \mathbf{n} \rangle$. Thus, vertices that are facing away from the camera receive a lower weighting, and self-occluded regions are discarded. In contrast to Van Rootseler et al. [vRSV12], who employ "classic" 3DMM fitting, our approach does not depend on the colour model or an illumination model fitting. Figure 5.7 shows an example image and the corresponding computed weighting, where red equals a 0° angle (facing the camera) and blue equals 90° or facing away from the camera.

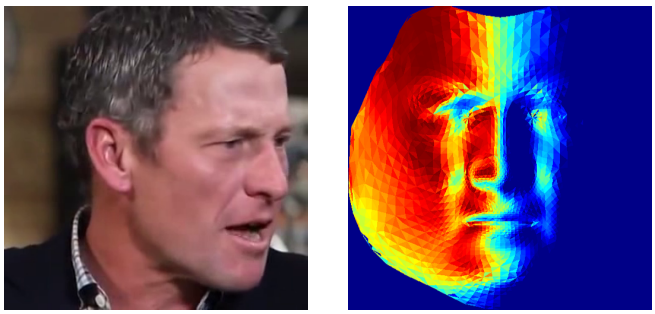


Figure 5.7: View visibility information (including regions of self-occlusions) from the 3D face model. *(left)*: Input frame. *(right)*: Per-vertex weighting. Using a JET colourmap, with red = 0° (facing the camera), and blue = 90° or facing away.

Given a video or a set of n images, we define the isomap with the merged texture values from all images to be $\mathcal{I} \in \mathbb{R}^{w \times h \times 3}$, where w and h are the width and height of the isomap, for example $w = h = 512$. $\mathcal{I}_{x,y}$ defines the triplet of colour values (r, g, b) at pixel location (x, y). To reconstruct the texture value $\mathcal{I}_{x,y}$ at each pixel location (x, y), we calculate a weighted average of all n images, each pixel weighed by its triangle's computed ω of a particular frame:

$$\mathcal{I}_{x,y} = \frac{1}{\sum_{i=1}^{n} \omega_{x,y}^{i}} \sum_{i=1}^{n} \omega_{x,y}^{i} c_{x,y}^{i}, \tag{5.4}$$

where $c_{x,y}^{i}$ is the colour in the isomap of frame i at location (x, y) and $\omega_{x,y}^{i}$ the weighting associated with the pixel's triangle.

In practice, this average can be computed very efficiently and thus used for real-time reconstruction. If run on a live video stream, the values of the current frame can be added to the previous average (with appropriate normalisation by the total number of frames), without having to recompute the values for all previous frames. Naturally, there is a trade-off between coverage and blurring with respect to the number of frames used, and thus on longer videos, averaging over all frames, even with a weighting, may not be particularly suitable. This will be briefly discussed in Section 7.1. While more complex fusion techniques could be applied, our method is particularly suited for real-time application and in that it allows the computation of an incremental texture model on a video stream, without having knowledge of the whole video in advance.

5.5 Performance Evaluation on In-the-wild Videos

In the following, we evaluate the quality of the texture reconstruction from in-the-wild video sequences. For this purpose, we choose the *300 Videos in the Wild* (300-VW) dataset ([SZC+15]). It contains videos more diverse than the ones from the KF-ITW dataset, and it includes scenarios such as

speeches and TV shows. The videos are between 1,000 and 3,000 frames in length. Most videos are fairly low-resolution, with the faces having an inter-eye-distance of 30 to 120 pixels, with a distance of around 60 pixels being the most common.

Since there is no 3D or texture ground truth available for these kinds of in-the-wild video datasets, we select 10 videos from the dataset which vary in subject identity, ethnicity, and age. For each of these 10 videos, we fit the 3DMM to a hand-selected left, frontal and right view that are annotated with accurate manual landmarks. We then create a reference isomap by manually merging the isomaps of the three view points. We then compare our fully automatic reconstruction with these reference isomaps.

We run the proposed fitting for each video on 68 automatically detected landmarks, which are obtained with the landmark detection of Huber et al. [HKC+17]. The texture is then reconstructed for each video with the approach presented in Section 5.4. Figure 5.8 shows an example frame from each video, the fully automatically reconstructed isomap, the reference isomap, and, purely for visualisation purposes, a rendering of the face in a novel pose. It can be seen that in general, the texture is reconstructed well, and the full face area is reconstructed, including the cheeks, side of the noses, and the regions near the ears, making full use of views of the subjects from larger pose angles. The accurate shape fitting results in no visible artefacts from the background of the videos in the isomap. The averaging over these many frames naturally results in slight blurring and loss of fine details.

5.6 Summary

This chapter presented the main contribution of extending the linear, closed-form solution shape fitting to multiple images. The overall proposed multi-

Figure 5.8: Texture fusion results on the 300-VW video database. *(First column)*: Frame from the original video. *(Second column)*: Reconstructed face texture using our view-based, weighted averaging. *(Third column)*: Ground truth face texture. *(Last column)*: Face rendering of a novel pose.

frame fitting exploits the fact that there is only one identity in a given video and is able to recover pose, identity and facial expressions. We demonstrated state-of-the-art results of the proposed algorithm on a newly released in-the-wild video dataset with 3D ground truth, where both pose variations and facial expressions are present. It was further shown that the proposed multi-frame landmark-fitting algorithm considerably outperforms "classic" 3DMM fitting, and we succeeded in showing that it is beneficial to use the temporal coherence of the subject identity within a video.

We further devised an efficient strategy to model face appearance from in-the-wild data with a weighted fusion of multiple images in texture space, and showed convincing results on a challenging in-the-wild video database with a fully automatic approach. In comparison to existing work, the proposed algorithm requires no subject-specific or manual training, reconstructs texture as well as dense 3D shape, and it is evaluated on a true in-the-wild video database. The main drawback of the texture fusion approach, the blur caused by fusing information from all frames of a given video, will be further discussed in Section 7.1.

Chapter 6

Conclusion

Over the last decade, there has been significant progress in 3D face reconstruction and 3D Morphable Model fitting from single images and monocular videos. Very recently, the reconstruction and fitting has even become possible to accomplish in real-time. However, many algorithms depend on a calibrated camera setup, subject-specific initialisation or training, or struggle or have not been evaluated on in-the-wild data which contains larger pose angles, facial expressions, and low-resolution. Most of the existing analysis-by-synthesis approaches are slow, complex, non-linear fitting algorithms, and fail to converge to a good optimum on in-the-wild data, and often rely heavily on a facial landmark constraint. Other algorithms are learning based, and suffer from the lack of in-the-wild training data with 3D ground truth.

2D facial landmark detection on the other hand has matured a lot over the recent years, and many algorithms can cope with in-the-wild images and a significant degree of pose and expression variations. It is much easier to obtain training data for 2D facial landmarks data, than for 3D.

Therefore, in this thesis, we proposed a real-time landmark fitting algorithm, crafted to deal with the challenges occurring on in-the-wild images. It consists of different steps to deal with facial expressions, 2D–3D contour correspondence mismatch, and large pose variations. Each of the compon-

ents contributes to making the final result a success, on a variety of images and databases, without any parameters to tune, or relying on a specific database to train on — except for the 2D facial landmark detector, which can be learned robustly from any of the huge variety of available 2D datasets. We demonstrated in this thesis that the proposed shape-to-landmarks fitting algorithm exhibits state of the art performance compared with the most recent nonlinear and learning-based methods, highlighting the difficulties of the fitting task and the shortcomings of nonlinear fitting methods.

In particular, the proposed algorithm consists of the following four components: *(1)* The closed-form solution shape identity fitting of Aldrian & Smith [AS13], *(2)* linear expression-blendshapes fitting, building on top of the shape identity fitting, *(3)* a dynamic approach to 2D facial contour fitting, and *(4)* a two-step closed-form approach to scaled orthographic camera estimation. Each of these components, and all components together as a whole, used in an iterative manner, are key to the successful and robust fitting to in-the-wild images. By using all the linear and closed-form terms, the proposed algorithm achieves a run time of up to 1000 frames per second on a regular Core i7 notebook CPU, and around 80 fps when the algorithm is run with the 3448 vertices model and for 5 iterations. Robustly recovering 3D shape with expressions from in-the-wild images at this speed is unprecedented in the literature, and opens doors to applications and research that have not been possible before.

We then extended the proposed approach to the case of fitting to multiple images, making use of the fact that there is one identity within a given video. The identity of a subject is recovered with a closed-form solution involving all images from a set or a video, and this step is alternated with recovering per-frame expression and camera parameters. We demonstrated the convergence of the algorithm and showed successfully that it is beneficial

to use the temporal coherence of the subject identity within a video. The proposed approach was evaluated on the recent KF-ITW dataset of in-the-wild video sequences, and we were able to closely match state-of-the-art performance with the proposed shape-to-landmarks fitting, compared with a nonlinear analysis-by-synthesis approach proposed in 2017.

Further, to reconstruct the appearance of a face, we combine the shape fitting with a simple weighted-mean based approach to fuse textures from multiple images. We apply it in a real-time context on uncalibrated, low-quality monocular in-the-wild videos, where the algorithm succeeds at reconstructing a shape and textural face representation, fusing different frames and view-angles. In comparison to other existing work, the proposed approach requires no subject-specific or manual training, and it was evaluated on true in-the-wild video datasets.

In summary, we evaluated the proposed approach quantitatively and qualitatively on a large battery of in-the-wild datasets: the AFLW2000-3D and HELEN image datasets, and the KF-ITW and 300-VW video datasets. Our proposed algorithm, reconstructing 3D shape only from landmarks, performs as good and better compared to two state-of-the-art fitting algorithms, while being much faster. The proposed algorithm is applicable to many domains, for example, fitting the model robustly to images and videos is a major milestone for face recognition applications. Because of its impressive run time, it can be applied to domains like Virtual Reality, where at minimum a refresh rate of 120 Hz is required. It is also very attractive for mobile devices, where power consumption is of utmost importance, and most often no GPU computing is available.

While the fact that we recover shape only from landmarks has been proven a strong advantage in this thesis, it is at the same time also a major drawback of the algorithm: inaccurately detected landmarks can not be

corrected or improved, and the geometry of the shape is in general limited by the shape defined by the 2D facial landmarks. In Chapter 7, we will discuss a number of possible research directions to keep the strong points of the proposed algorithm while improving on some of its weaknesses.

Finally, the research in this thesis was published as a lightweight open source library for real-time 3D morphable face model fitting. The library is one of the first and one of very few public and maintained 3DMM fitting frameworks. We have also made the 3448 vertices 3D shape model and expression blendshapes available as *Surrey Face Model* as part of the software. By making our software available, we close a significant gap in the research landscape of 3DMM fitting algorithms, and have already succeeded at making 3D face models easier to use and more widely available, to the benefit of the whole research community.

Chapter 7

Future Work

In this chapter, we discuss directions for future work, and present preliminary research and experiments that have been conducted.

First, the texture merged with the weighted mean based method presented in Section 5.4 exhibits a noteworthy amount of blur from the averaging, particularly if the averaging is done over a whole video. This can be amended by using a fusion technique that causes less blur on the one hand, and selecting a limited number of good frames for the fusion on the other hand. We propose an approach covering both in Section 7.1, and present preliminary results.

Second, since the shape in the proposed fitting algorithm is only fitted to landmarks, the obtained geometry is limited by the shape defined by the 2D facial landmarks. Even though one solution is to use more landmarks, it would still not be possible to model for example wrinkles or the exact nose shape of a particular subject. Also, if any or several of the landmarks are detected imprecisely, that mistake cannot be corrected by the fitting. Hence, some form of image-related cost function or shape from shading would be needed to further improve the shape accuracy. However, at the same time, we want to avoid all the drawbacks of the complex non-linear fitting methods. Section 7.2 presents a possible research direction together with ideas for future work.

7.1 Super-Resolution Texture Fusion

To combat the shortcomings of the texture fusion method presented in Sections 5.4 and 5.5, we conduct preliminary research in two directions: First, we devise a strategy to qualitatively rate a frame, and then select frames where the quality measure is highest. The selected frames are additionally placed in designated bins, separated by the yaw pose angle. Second, using these selected key frames, we propose to use a median-based super-resolution approach that can be employed in real-time. In the following, we briefly outline the proposed technique and show preliminary results.

7.1.1 Key Frame Selection

For each frame of a video, we rate the face region with regards to its image quality and suitability to use for texture reconstruction. Given the facial landmarks of a frame, we first extract a patch containing only the face by using the bounding box enclosing the landmarks. We rate this patch using the variance of Laplacian measure by Pech-Pacheco et al. [PCCF00], which is a measure for focus or sharpness of an image. Additionally, we expect frames with fewer facial expressions to have less deformations in the texture, which would be beneficial for the fusion. We thus penalise frames with a large norm of the blendshape coefficients, $\|\boldsymbol{\psi}\|$, where $\boldsymbol{\psi}$ is estimated using a rough initial pose and expression fit using only inner landmarks (see the first two steps of Figure 4.7). Each frame is then associated with a score s:

$$s = L\frac{\tau}{\|\boldsymbol{\psi}\|}\,, \tag{7.1}$$

where L is the variance of Laplacian measure and τ a parameter to influence how much stronger expressions are penalised.

Because we would like to capture a face from as many different viewpoints as possible, we then divide the $\pm 90°$ yaw pose space into bins of $20°$ intervals.

Each bin can contain a maximum of p key frames. A frame is used as key frame if the bin corresponding to its particular pose is not full, or, if any of the existing key frames of that pose bin have a lower score than the current frame. In the latter case, the currently present key frame is replaced with the newer one with the higher score. For example, if $p = 1$, only the very best frame for each pose bin, according to the score s, is kept. Figure 7.1 shows the resulting key frames after running the algorithm over a whole video of 1500 frames from the 300-VW dataset. We can observe that the chosen frames appear relatively sharp, without motion blur (some frames of the videos contain a notable amount of blurring), and in relatively neutral expression (mainly the mouth is closed). The pose bin at the very left side of the figure (-90 to $-70°$ yaw) as well as the two bins on the very right side are empty, since no frame in the video contained the subject in such a strong pose.

Figure 7.1: Top-scoring key frame for each pose bin for an example video. The bins range from -90° yaw to +90° in a 20° interval.

7.1.2 Median-based Super-Resolution

Using these selected key frames, we propose to use a super-resolution approach that can be employed in real-time. In 2015, Maier et al. [MSC15] proposed an approach to texture mapping of 3D models using super-resolution key frames. While their method uses depth data from an RGB-D camera, in our case, the input is a set of 2D key frames captured from a monocular camera, and the 3D face model fitted to the landmarks of these images.

In an initial setup, we proceed as follows:

- The texture of each key frame is remapped to a common reference tex-

ture map of higher resolution using the dense correspondence obtained from the model fitting.

- For each pixel of the final texture map, a weighted median over all key frames is computed. The final colour value $\hat{c}$ of a pixel is given by:

$$\hat{c} = \arg\min_{c} \sum_{(c_i,\omega_i)\in\mathcal{O}} \omega_i \left\| c - c_i \right\| \tag{7.2}$$

  where $\mathcal{O}$ is the set of all colour values and weights of all key frames for a particular pixel.

- The weight ω for a pixel consists of the previously calculated frame score and a per-vertex weighting based on the view-angle of the vertex w.r.t. the camera's viewing direction:

$$\omega = \frac{\langle \mathbf{d}, \mathbf{n} \rangle \, L \, \tau}{\|\boldsymbol{\psi}\|} \tag{7.3}$$

  where $\mathbf{d}$ is the camera viewing direction and $\mathbf{n}$ the normal of the vertex.

- The texture map is recomputed whenever a new key frame is added.

Figure 7.2 depicts an overview of the proposed approach, with the 3D shape fitting and key frame selection first, followed by the median-based super-resolution texture fusion. The resulting super-resolved texture map looks more crisp and contains more details than the one obtained in Section 5.5 by averaging over the whole video (see Figure 5.8, row 5). However, in practice, we found that the method generates visible stitching artefacts on many of the videos, and that further investigation is necessary. With the averaging of Section 5.4, these artefacts do not appear, as the averaging over the whole video acts as a smoothing, but at the cost of lost texture details. Maier et al. [MSC15], in their original approach, are able to bring the key frames into better correspondence with the help of depth information, and through the depth information they are also able

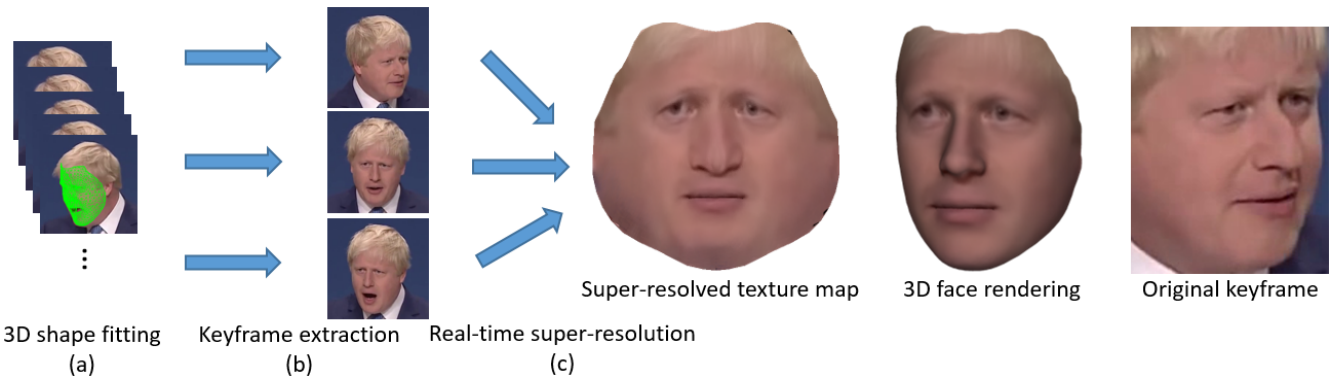


Figure 7.2: Prototype of the proposed real-time super-resolution texture-fusion approach. (a) 3D shape fitting in each video frame. (b) Key frame extraction based on pose and frame quality. (c) Median-based super-resolution fusion of the key frames. The inter-eye-distance of the input frames is around 45 pixels.

to recover 3D shape that is in better correspondence amongst frames than the result of our fitting. Other approaches that employ similar texture fusion approaches, like for example the multi-view reconstruction method by Starck and Hilton [SH03], work with calibrated camera setups, often in studios, where these texture fusion approaches work very well.

One challenge is thus to devise a method that works robustly on in-the-wild videos from monocular video streams, in an uncalibrated setup. We think that the research direction sketched in this section can lay the ground work, and two directions are worth further investigating: First, the texture maps could be post-processed after the fitting and brought into better dense correspondence with an optical flow based method, like for example proposed by Volino et al. [VCCH14]. In their work, they correct for texture misalignment due to geometric and calibration errors, which in our case, would be inaccuracies from the shape fitting. The second research direction is to improve the shape fitting itself, where the next section (Section 7.2) outlines a possible way.

7.2 Illumination-invariant Appearance Model

In the following, we present a possible way of improving the accuracy of the shape fitting. The fact that, in the proposed shape fitting, the geometry is limited by the face shape defined by the 2D facial landmarks, and that imprecisely detected landmarks cannot be corrected, is an inherent limitation of the algorithm. An obvious suggestion, using a landmark detector that outputs a larger number of facial landmarks, does not solve the core of the problem. However, we also want to avoid using the traditional 3DMM albedo model and an illumination model, as we have showed throughout this thesis that the classical 3DMM fitting approach and nonlinear methods struggle with fitting to in-the-wild images. One promising solution that we see is the use of an illumination-invariant albedo model. The idea was first proposed by Hu et al. [HCY+14], who present what they call an *Albedo Based 3D Morphable Model* (AB3DMM). They apply an illumination normalisation technique like for example *Single scale retinex* ([RJW96]) or *gradientfaces* ([ZTF+09]) to the photographs captured by the 3D face scanner, and then build a PCA model of face appearance in one of these illumination-normalised spaces, instead of the traditionally used RGB space. The hope is to make the optimisation problem easier to solve by not incorporating an explicit illumination model (like e.g. the Phong model) into the cost function. When the model is fitted to an input image, the same illumination normalisation technique is first applied to the image, and then, they use the traditional analysis-by-synthesis approach of solving for camera, shape and (illumination-invariant) appearance parameters with a nonlinear cost function containing a term measuring the error between model projection and input image in that illumination-normalised space. While Hu et al. showed promising results in controlled conditions, their approach still relies on nonlinear optimisation, and has a run time of the

order of a minute.

We believe that, with improvements, an approach like theirs in general could be the ideal bridge between a landmarks-only based fitting approach, and the traditional analysis-by-synthesis approach. Figure 7.3 shows a sample of such a learned illumination-invariant appearance model. In this case, we learned the PCA model on the Laplacian of all subject's isomaps. Since the Laplacian consists of real values, we rescale the values in the figure for visualisation purposes. In contrast to Hu et al., we apply the illumination normalisation to the isomaps after the scans have already been registered.

Figure 7.3: A sample of a model learned on the Laplacian of all subject's isomaps. The Laplacian values have been rescaled to visualise the model, where black indicates negative values and white positive values.

An open research question is how to subsequently fit the model to novel images, while avoiding the drawbacks of existing nonlinear fitting methods. One interesting approach is the *Linear Shape and Texture* (LiST) fitting algorithm from Romdhani et al. [RBV02]. They propose to use linear solutions to shape and texture parameter estimation, together with applying optical flow between the input image and the rendered model image, to improve the shape recovery. They present promising results on the PIE database, however, like most 3DMM fitting algorithms, are not able to fit facial expressions and lack an evaluation on unconstrained in-the-wild

images.

The recently presented ITW-fitting approach by Booth et al. [BAP+17a], which we compared our approach against in Section 5.2, also has similarity with the presented illumination-invariant appearance model, with promising results on in-the-wild images and videos. Their "appearance" model is based on SIFT image features, and learned from in-the-wild 2D images. However in our case we would still learn the illumination invariant appearance model from 3D scans, to leverage the dense correspondence throughout the whole face region.

Appendix A

Derivation of the Closed-form Shape Fitting Solution

The PCA identity shape fitting in Section 4.2 is expressed in terms of a regularised quadratic form which has a closed form solution. In the following, we give the derivation, taken from Aldrian & Smith [AS13].

Expanding the regularised quadratic form gives:

$$\begin{aligned}
\mathbb{E} &= (\mathbf{A}\mathbf{x}+\mathbf{b})^{\mathrm{T}}\boldsymbol{\Omega}(\mathbf{A}\mathbf{x}+\mathbf{b}) + \lambda\left\|\mathbf{x}\right\|^2 \\
&= \left[(\mathbf{A}\mathbf{x})^{\mathrm{T}}\boldsymbol{\Omega} + \mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}\right](\mathbf{A}\mathbf{x}+\mathbf{b}) + \lambda\left\|\mathbf{x}\right\|^2 \\
&= (\mathbf{A}\mathbf{x})^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A}\mathbf{x} + (\mathbf{A}\mathbf{x})^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{b} + \mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A}\mathbf{x} + \mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{b} + \lambda\left\|\mathbf{x}\right\|^2 \qquad \text{(A.1)} \\
&= \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A}\mathbf{x} + \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{b} + \mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A}\mathbf{x} + \mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{b} + \lambda\left\|\mathbf{x}\right\|^2 \\
&= \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A}\mathbf{x} + (\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{b})^{\mathrm{T}}\mathbf{x} + \mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A}\mathbf{x} + \mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{b} + \lambda\left\|\mathbf{x}\right\|^2 .
\end{aligned}$$

The solution with respect to $\mathbf{x}$ is given by:

$$\begin{aligned}
\frac{\mathrm{d}\mathbb{E}}{\mathrm{d}\mathbf{x}} &= 2\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A} + (\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{b})^{\mathrm{T}} + \mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A} + 2\lambda\mathbf{x}^{\mathrm{T}} &&= 0 \\
&= 2\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A} + \mathbf{b}^{\mathrm{T}}(\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega})^{\mathrm{T}} + \mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A} + 2\lambda\mathbf{x}^{\mathrm{T}} \\
&= 2\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A} + \mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}^{\mathrm{T}}\mathbf{A} + \mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A} + 2\lambda\mathbf{x}^{\mathrm{T}} \\
&= 2\mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A} + 2\mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A} + 2\lambda\mathbf{x}^{\mathrm{T}} \qquad \text{(A.2)} \\
&= \mathbf{x}^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A} + \lambda\mathbf{x}^{\mathrm{T}} + \mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A} \\
&= (\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A})^{\mathrm{T}}\mathbf{x} + \lambda\mathbf{x} + (\mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A})^{\mathrm{T}} \\
&= \mathbf{A}^{\mathrm{T}}(\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega})^{\mathrm{T}}\mathbf{x} + \lambda\mathbf{x} + \mathbf{A}^{\mathrm{T}}(\mathbf{b}^{\mathrm{T}}\boldsymbol{\Omega})^{\mathrm{T}}
\end{aligned}$$

$$
\begin{aligned}
&= (\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A} + \lambda\mathbf{I})\mathbf{x} + \mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}^{\mathrm{T}}\mathbf{b} \\
\mathbf{x} &= -(\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}\mathbf{A} + \lambda\mathbf{I})^{-1}(\mathbf{A}^{\mathrm{T}}\boldsymbol{\Omega}^{\mathrm{T}}\mathbf{b}).
\end{aligned}
$$

The solution to the expression fitting is derived analogously, but simplifies, since $\boldsymbol{\Omega}$ is not used in that term (see Section 4.3).

Bibliography

[AAB+14] Joan Alabort-i-Medina, Epameinondas Antonakos, James Booth, Patrick Snape, and Stefanos Zafeiriou. Menpo: A comprehensive platform for parametric image alignment and visual deformable models. In Kien A. Hua, Yong Rui, Ralf Steinmetz, Alan Hanjalic, Apostol Natsev, and Wenwu Zhu, editors, *Proceedings of the ACM International Conference on Multimedia, MM '14, Orlando, FL, USA, November 03 - 07, 2014*, pages 679–682. ACM, 2014.

[AKV08] Brian Amberg, Reinhard Knothe, and Thomas Vetter. Expression invariant 3d face recognition with a morphable model. In *8th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2008), Amsterdam, The Netherlands, 17-19 September 2008*, pages 1–6. IEEE Computer Society, 2008.

[Amb11] Brian Amberg. *Editing faces in videos*. PhD thesis, University of Basel, 2011.

[AMO] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. `http://ceres-solver.org`.

[ARV07] Brian Amberg, Sami Romdhani, and Thomas Vetter. Optimal step nonrigid ICP algorithms for surface registration. In

2007 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007), 18-23 June 2007, Minneapolis, Minnesota, USA. IEEE Computer Society, 2007.

[AS13] Oswald Aldrian and William A. P. Smith. Inverse rendering of faces with a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(5):1080–1093, 2013.

[BAP+17a] James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, and Stefanos Zafeiriou. 3d face morphable models "in-the-wild". *CoRR*, abs/1701.05360, 2017.

[BAP+17b] James Booth, Epameinondas Antonakos, Stylianos Ploumpis, George Trigeorgis, Yannis Panagakis, and Stefanos Zafeiriou. 3d face morphable models "in-the-wild". In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, 2017.

[BBPV03] Volker Blanz, Curzio Basso, Tomaso A. Poggio, and Thomas Vetter. Reanimating faces in images and video. *Comput. Graph. Forum*, 22(3):641–650, 2003.

[BHB00] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3d shape from image streams. In *2000 Conference on Computer Vision and Pattern Recognition (CVPR 2000), 13-15 June 2000, Hilton Head, SC, USA*, pages 2690–2696. IEEE Computer Society, 2000.

[Bra01] Matthew Brand. Morphable 3d models from video. In *2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001), with CD-ROM,*

8-14 December 2001, Kauai, HI, USA, pages 456–463. IEEE Computer Society, 2001.

[BRZ+16] James Booth, Anastasios Roussos, Stefanos Zafeiriou, Allan Ponniahy, and David Dunaway. A 3d morphable model learnt from 10,000 faces. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5543–5552. IEEE Computer Society, 2016.

[BSBW14] Alan Brunton, Augusto Salazar, Timo Bolkart, and Stefanie Wuhrer. Review of statistical shape spaces for 3d data with comparative analysis for human faces. *Computer Vision and Image Understanding*, 128:1–17, 2014.

[BSBW16] Anil Bas, William A. P. Smith, Timo Bolkart, and Stefanie Wuhrer. Fitting a 3d morphable model to edges: A comparison between hard and soft correspondences. *CoRR*, abs/1602.01125, 2016.

[BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In Warren N. Waggenspack, editor, *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1999, Los Angeles, CA, USA, August 8-13, 1999*, pages 187–194. ACM, 1999.

[BW15] Timo Bolkart and Stefanie Wuhrer. 3d faces in motion: Fully automatic registration and statistical analysis. *Computer Vision and Image Understanding*, 131:100–115, 2015.

[BW16] Timo Bolkart and Stefanie Wuhrer. A robust multilinear model learning framework for 3d faces. In *2016 IEEE Conference on*

Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 4911–4919. IEEE Computer Society, 2016.

[BWP13] Sofien Bouaziz, Yangang Wang, and Mark Pauly. Online modeling for realtime facial animation. *ACM Trans. Graph.*, 32(4):40:1–40:10, 2013.

[BZD+15] J. Ross Beveridge, Hao Zhang, Bruce A. Draper, Patrick J. Flynn, Zhen-Hua Feng, Patrik Huber, Josef Kittler, Zhiwu Huang, Shaoxin Li, Yan Li, Meina Kan, Ruiping Wang, Shiguang Shan, Xilin Chen, Haoxiang Li, Gang Hua, Vitomir Struc, Janez Krizaj, Changxing Ding, Dacheng Tao, and P. Jonathon Phillips. Report on the FG 2015 video person recognition evaluation. In *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2015, Ljubljana, Slovenia, May 4-8, 2015*, pages 1–8. IEEE Computer Society, 2015.

[CBZB15] Chen Cao, Derek Bradley, Kun Zhou, and Thabo Beeler. Real-time high-fidelity facial performance capture. *ACM Trans. Graph.*, 34(4):46, 2015.

[CET98] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. In Hans Burkhardt and Bernd Neumann, editors, *Computer Vision - ECCV'98, 5th European Conference on Computer Vision, Freiburg, Germany, June 2-6, 1998, Proceedings, Volume II*, volume 1407 of *Lecture Notes in Computer Science*, pages 484–498. Springer, 1998.

[CET01] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 23(6):681–685, 2001.

[CHZ14] Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. Graph.*, 33(4):43:1–43:10, 2014.

[CT92] Timothy F. Cootes and Christopher J. Taylor. Active shape models - 'smart snakes'. In David C. Hogg and Roger Boyle, editors, *Proceedings of the British Machine Vision Conference, BMVC 1992, Leeds, UK, September, 1992*, pages 1–10. BMVA Press, 1992.

[CTCG95] Timothy F. Cootes, Christopher J. Taylor, David H. Cooper, and Jim Graham. Active shape models-their training and application. *Computer Vision and Image Understanding*, 61(1):38–59, 1995.

[CWLZ13] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 3d shape regression for real-time facial animation. *ACM Trans. Graph.*, 32(4):41:1–41:10, 2013.

[CWZ+14] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3d facial expression database for visual computing. *IEEE Trans. Vis. Comput. Graph.*, 20(3):413–425, 2014.

[EF78] Paul Ekman and Wallace V. Friesen. *Facial Action Coding System: A Technique for the Measurement of Facial Movement.* Consulting Psychologists Press, Palo Alto, 1978.

[EFH02] Paul Ekman, Wallace V. Friesen, and Joseph C. Hager. Facial action coding system: The manual on CD ROM, 2002.

[Ekm89] Paul Ekman. The argument and evidence about universals in facial expressions of emotion. In Hugh Ed Wagner and Antony Ed Manstead, editors, *Handbook of social psychophysiology*, pages 143–164. John Wiley & Sons, 1989.

[Ekm92] Paul Ekman. An argument for basic emotions. *Cognition and Emotion*, 6(3-4):169–200, 1992.

[ESB+16] Bernhard Egger, Andreas Schneider, Clemens Blumer, Andreas Forster, Sandro Schönborn, and Thomas Vetter. Occlusion-aware 3d morphable face models. In *Proceedings of the British Machine Vision Conference, BMVC 2016, York, UK, September, 2016*. BMVA Press, 2016.

[ESFV14] Bernhard Egger, Sandro Schönborn, Andreas Forster, and Thomas Vetter. Pose normalization for eye gaze estimation and facial attribute description from still images. In Xiaoyi Jiang, Joachim Hornegger, and Reinhard Koch, editors, *Pattern Recognition - 36th German Conference, GCPR 2014, Münster, Germany, September 2-5, 2014, Proceedings*, volume 8753 of *Lecture Notes in Computer Science*, pages 317–327. Springer, 2014.

[FHK+15] Zhen-Hua Feng, Patrik Huber, Josef Kittler, William J. Christmas, and Xiaojun Wu. Random cascaded-regression copse for robust facial landmark detection. *IEEE Signal Process. Lett.*, 22(1):76–80, 2015.

[FKC+17] Zhen-Hua Feng, Josef Kittler, William Christmas, Patrik Huber, and Xiao-Jun Wu. Dynamic attention-controlled cas-

caded shape regression exploiting training data augmentation and fuzzy-set sample weighting. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*. IEEE Computer Society, 2017.

[GJ+10] Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

[GVWT13] Pablo Garrido, Levi Valgaerts, Chenglei Wu, and Christian Theobalt. Reconstructing detailed dynamic face geometry from monocular video. *ACM Trans. Graph.*, 32(6):158:1–158:10, 2013.

[HCH+16] Patrik Huber, William J. Christmas, Adrian Hilton, Josef Kittler, and Matthias Rätsch. Real-time 3d face super-resolution from monocular in-the-wild videos. In *Special Interest Group on Computer Graphics and Interactive Techniques Conference, SIGGRAPH '16, Anaheim, CA, USA, July 24-28, 2016, Posters Proceedings*, pages 67:1–67:2. ACM, 2016.

[HCY+14] Guosheng Hu, Chi-Ho Chan, Fei Yan, William J. Christmas, and Josef Kittler. Robust face recognition by an albedo based 3d morphable model. In *IEEE International Joint Conference on Biometrics, Clearwater, IJCB 2014, FL, USA, September 29 - October 2, 2014*, pages 1–8. IEEE, 2014.

[HDG+12] Catherine Herold, Vincent Despiegel, Stéphane Gentric, Séverine Dubuisson, and Isabelle Bloch. Head shape estimation using a particle filter including unknown static parameters. In Gabriela Csurka and José Braz, editors, *VISAPP 2012 -*

Proceedings of the International Conference on Computer Vision Theory and Applications, Volume 2, Rome, Italy, 24-26 February, 2012., pages 284–293. SciTePress, 2012.

[HDG+14] Catherine Herold, Vincent Despiegel, Stéphane Gentric, Séverine Dubuisson, and Isabelle Bloch. Recursive head reconstruction from multi-view video sequences. *Computer Vision and Image Understanding*, 122:182–201, 2014.

[HFC+15] Patrik Huber, Zhen-Hua Feng, William J. Christmas, Josef Kittler, and Matthias Rätsch. Fitting 3d morphable face models using local features. In *2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, September 27-30, 2015*, pages 1195–1199. IEEE, 2015.

[HHT+16] Patrik Huber, Guosheng Hu, Jose Rafael Tena, Pouria Mortazavian, Willem P. Koppen, William J. Christmas, Matthias Rätsch, and Josef Kittler. A multiresolution 3d morphable face model and fitting framework. In Nadia Magnenat-Thalmann, Paul Richard, Lars Linsen, Alexandru Telea, Sebastiano Battiato, Francisco H. Imai, and José Braz, editors, *Proceedings of the 11th Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2016) - Volume 4: VISAPP, Rome, Italy, February 27-29, 2016.*, pages 79–86. SciTePress, 2016.

[HKC+17] Patrik Huber, Philipp Kopp, William J. Christmas, Matthias Rätsch, and Josef Kittler. Real-time 3d face fitting and texture fusion on in-the-wild videos. *IEEE Signal Process. Lett.*, 24(4):437–441, 2017.

[HYK+17] Guosheng Hu, Fei Yan, Josef Kittler, William J. Christmas, Chi-Ho Chan, Zhen-Hua Feng, and Patrik Huber. Efficient 3d morphable face model fitting. *Pattern Recognition*, 67:366–379, 2017.

[HZ03] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, New York, NY, USA, second edition, 2003.

[IBP15] Alexandru Eugen Ichim, Sofien Bouaziz, and Mark Pauly. Dynamic 3d avatar creation from hand-held video input. *ACM Trans. Graph.*, 34(4):45, 2015.

[JCK15] László A. Jeni, Jeffrey F. Cohn, and Takeo Kanade. Dense 3d face alignment from 2d videos in real-time. In *11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, FG 2015, Ljubljana, Slovenia, May 4-8, 2015*, pages 1–8. IEEE Computer Society, 2015.

[KHF+16] Josef Kittler, Patrik Huber, Zhen-Hua Feng, Guosheng Hu, and William J. Christmas. 3d morphable face models and their applications. In Francisco José Perales López and Josef Kittler, editors, *Articulated Motion and Deformable Objects - 9th International Conference, AMDO 2016, Palma de Mallorca, Spain, July 13-15, 2016, Proceedings*, volume 9756 of *Lecture Notes in Computer Science*, pages 185–206. Springer, 2016.

[Kno09] Reinhard Knothe. *A global-to-local model for the representation of human faces*. PhD thesis, University of Basel, 2009.

[KWRB11] Martin Köstinger, Paul Wohlhart, Peter M. Roth, and Horst Bischof. Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization. In

IEEE International Conference on Computer Vision Workshops, ICCV 2011 Workshops, Barcelona, Spain, November 6-13, 2011, pages 2144–2151. IEEE, 2011.

[LBL+12] Vuong Le, Jonathan Brandt, Zhe Lin, Lubomir D. Bourdev, and Thomas S. Huang. Interactive facial feature localization. In Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision - ECCV 2012 - 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part III*, volume 7574 of *Lecture Notes in Computer Science*, pages 679–692. Springer, 2012.

[LH95] C. Lawson and R. Hanson. *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics, 1995.

[ML14] Marius Muja and David G. Lowe. Scalable nearest neighbor algorithms for high dimensional data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(11):2227–2240, 2014.

[MSC15] Robert Maier, Jörg Stückler, and Daniel Cremers. Super-resolution keyframe fusion for 3d modeling with high-quality textures. In Michael S. Brown, Jana Kosecká, and Christian Theobalt, editors, *2015 International Conference on 3D Vision, 3DV 2015, Lyon, France, October 19-22, 2015*, pages 536–544. IEEE Computer Society, 2015.

[Mur12] Michael Muré. Face recognition and animation software. Technical report, Multimedia Signal Processing Lab, Ajou University, Suwon, Korea, 2012.

[MXB07] Iain A. Matthews, Jing Xiao, and Simon Baker. 2d vs. 3d deformable face models: Representational power, construction,

and real-time fitting. *International Journal of Computer Vision*, 75(1):93–113, 2007.

[NIH+11] Richard A. Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J. Davison, Pushmeet Kohli, Jamie Shotton, Steve Hodges, and Andrew W. Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *10th IEEE International Symposium on Mixed and Augmented Reality, ISMAR 2011, Basel, Switzerland, October 26-29, 2011*, pages 127–136. IEEE Computer Society, 2011.

[PCCF00] José Luis Pech-Pacheco, Gabriel Cristóbal, Jesús Chamorro-Martínez, and Joaquín Fernández-Valdivia. Diatom autofocusing in brightfield microscopy: a comparative study. In *15th International Conference on Pattern Recognition, ICPR'00, Barcelona, Spain, September 3-8, 2000.*, pages 3318–3321. IEEE Computer Society, 2000.

[PKA+09] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In Stefano Tubaro and Jean-Luc Dugelay, editors, *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS 2009, 2-4 September 2009, Genova, Italy*, pages 296–301. IEEE Computer Society, 2009.

[PL06] Frederic Pighin and J. P. Lewis. Performance-driven facial animation. In *ACM SIGGRAPH 2006 Courses*, SIGGRAPH '06, New York, NY, USA, 2006. ACM.

[RBV02] Sami Romdhani, Volker Blanz, and Thomas Vetter. Face identification by fitting a 3d morphable model using linear

shape and texture error functions. In Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, editors, *Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part IV*, volume 2353 of *Lecture Notes in Computer Science*, pages 3–19. Springer, 2002.

[RJW96] Zia-ur Rahman, Daniel J. Jobson, and Glenn A. Woodell. Multi-scale retinex for color image enhancement. In *Proceedings 1996 International Conference on Image Processing, Lausanne, Switzerland, September 16-19, 1996*, pages 1003–1006. IEEE Computer Society, 1996.

[Rod07] J. R. Tena Rodríguez. *3D Face Modelling for 2D+3D Face Recognition*. PhD thesis, University of Surrey, 2007.

[RV05] Sami Romdhani and Thomas Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005), 20-26 June 2005, San Diego, CA, USA*, pages 986–993. IEEE Computer Society, 2005.

[SAD+08] Arman Savran, Nese Alyüz, Hamdi Dibeklioglu, Oya Çeliktutan, Berk Gökberk, Bülent Sankur, and Lale Akarun. Bosphorus database for 3d face analysis. In Ben A. M. Schouten, Niels Christian Juul, Andrzej Drygajlo, and Massimo Tistarelli, editors, *Biometrics and Identity Management, First European Workshop, BIOID 2008, Roskilde, Denmark, May 7-9, 2008. Revised Selected Papers*, volume 5372 of *Lecture Notes in Computer Science*, pages 47–56. Springer, 2008.

[SAT+16] Christos Sagonas, Epameinondas Antonakos, Georgios Tzimiropoulos, Stefanos Zafeiriou, and Maja Pantic. 300 faces in-the-wild challenge: database and results. *Image Vision Comput.*, 47:3–18, 2016.

[SCD+06] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006), 17-22 June 2006, New York, NY, USA*, pages 519–528. IEEE Computer Society, 2006.

[SEFV15] Sandro Schönborn, Bernhard Egger, Andreas Forster, and Thomas Vetter. Background modeling for generative image models. *Computer Vision and Image Understanding*, 136:117–127, 2015.

[SEMFV16] Sandro Schönborn, Bernhard Egger, Andreas Morel-Forster, and Thomas Vetter. Markov chain monte carlo for automated face image analysis. *International Journal of Computer Vision*, pages 1–24, 2016.

[SFEV13] Sandro Schönborn, Andreas Forster, Bernhard Egger, and Thomas Vetter. A monte carlo strategy to integrate detection and model-based face analysis. In *Pattern Recognition - 35th German Conference, GCPR 2013, Saarbrücken, Germany, September 3-6, 2013. Proceedings*, volume 8142 of *Lecture Notes in Computer Science*, pages 101–110. Springer, 2013.

[SH03] Jonathan Starck and Adrian Hilton. Model-based multiple view reconstruction of people. In *9th IEEE International Conference*

on Computer Vision (ICCV 2003), 14-17 October 2003, Nice, France, pages 915–922. IEEE Computer Society, 2003.

[Smi16] William A. P. Smith. *The Perspective Face Shape Ambiguity*, pages 299–319. Springer International Publishing, Cham, 2016.

[SSS08] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[SZC+15] Jie Shen, Stefanos Zafeiriou, Grigoris G. Chrysos, Jean Kossaifi, Georgios Tzimiropoulos, and Maja Pantic. The first facial landmark tracking in-the-wild challenge: Benchmark and results. In *2015 IEEE International Conference on Computer Vision Workshop, ICCV Workshops 2015, Santiago, Chile, December 7-13, 2015*, pages 1003–1011. IEEE, 2015.

[THB03] Lorenzo Torresani, Aaron Hertzmann, and Christoph Bregler. Learning non-rigid 3d shape from 2d motion. In Sebastian Thrun, Lawrence K. Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16 [Neural Information Processing Systems, NIPS 2003, December 8-13, 2003, Vancouver and Whistler, British Columbia, Canada]*, pages 1555–1562. MIT Press, 2003.

[THHI06] Jose Rafael Tena, Miroslav Hamouz, Adrian Hilton, and John Illingworth. A validated method for dense non-rigid 3d face registration. In *Advanced Video and Signal Based Surveillance, 2006 IEEE International Conference on Video and Signal Based Surveillance (AVSS'06), 22-24 November 2006, Sydney, Australia.*, page 81. IEEE Computer Society, 2006.

[TMHF99] Bill Triggs, Philip F. McLauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle adjustment - A modern synthesis. In Bill Triggs, Andrew Zisserman, and Richard Szeliski, editors, *Vision Algorithms: Theory and Practice, International Workshop on Vision Algorithms, held during ICCV '99, Corfu, Greece, September 21-22, 1999, Proceedings*, volume 1883 of *Lecture Notes in Computer Science*, pages 298–372. Springer, 1999.

[TP91] Matthew A. Turk and Alex Pentland. Face recognition using eigenfaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 1991, 3-6 June, 1991, Lahaina, Maui, Hawaii, USA*, pages 586–591. IEEE, 1991.

[TSL00] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

[TZS+16] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Nießner. Face2face: Real-time face capture and reenactment of rgb videos. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2016.

[VCCH14] Marco Volino, Dan Casas, John P. Collomosse, and Adrian Hilton. Optimal representation of multiple view video. In Michel François Valstar, Andrew P. French, and Tony P. Pridmore, editors, *British Machine Vision Conference, BMVC 2014, Nottingham, UK, September 1-5, 2014*. BMVA Press, 2014.

[vRSV11] R.T.A. van Rootseler, L.J. Spreeuwers, and R.N.J. Veldhuis. Application of 3d morphable models to faces in video images.

In *Proceedings of the 32nd Symposium on Information Theory in the Benelux (WIC11)*, May 2011.

[vRSV12] R.T.A. van Rootseler, L.J. Spreeuwers, and R.N.J. Veldhuis. Using 3d morphable models for face recognition in video. In *Proceedings of the 33rd WIC Symposium on Information Theory in the Benelux*, pages 235–242, Enschede, the Netherlands, May 2012. Werkgemeenschap voor Informatie- en Communicatietheorie, WIC.

[VZ01] Luiz Velho and Denis Zorin. 4-8 subdivision. *Computer Aided Geometric Design*, 18(5):397–427, 2001.

[WBGB16] Chenglei Wu, Derek Bradley, Markus H. Gross, and Thabo Beeler. An anatomically-constrained local deformation model for monocular face capture. *ACM Trans. Graph.*, 35(4):115, 2016.

[WBLP11] Thibaut Weise, Sofien Bouaziz, Hao Li, and Mark Pauly. Real-time performance-based facial animation. *ACM Trans. Graph.*, 30(4):77:1–77:10, 2011.

[XBMK04] Jing Xiao, Simon Baker, Iain A. Matthews, and Takeo Kanade. Real-time combined 2d+3d active appearance models. In *2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004), with CD-ROM, 27 June - 2 July 2004, Washington, DC, USA*, pages 535–542. IEEE Computer Society, 2004.

[YWS+06] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J. Rosato. A 3d facial expression database for facial behavior research. In *Seventh IEEE International Conference on Automatic Face and Gesture Recognition (FG 2006), 10-12 April*

2006, Southampton, UK, pages 211–216. IEEE Computer Society, 2006.

[ZLL+16] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3d solution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 146–155. IEEE Computer Society, 2016.

[ZTF+09] Taiping Zhang, Yuan Yan Tang, Bin Fang, Zhaowei Shang, and Xiaoyu Liu. Face recognition under varying illumination using gradientfaces. *IEEE Trans. Image Processing*, 18(11):2599–2606, 2009.

[ZYY+15] Xiangyu Zhu, Junjie Yan, Dong Yi, Zhen Lei, and Stan Z. Li. Discriminative 3d morphable model fitting. In *FG*, pages 1–8. IEEE Computer Society, 2015.